

---

Nazwa Projektu

projektrwd

---

Typ dokumentu

HTML5 z JS (React), SASS (użyłem SCSS)

---

Kodowanie znaków

utf-8

---

Tytuły stron

Bonsai.pl:

- Home - Bonsai.pl
- Galeria - Bonsai.pl
- Rodzaje - Bonsai.pl

---

Budowa strony

Zmienne Globalne:.....	4
Kolory podstawowe.....	4
Kolory tekstu.....	4
Cień .....	5
Media Queries .....	5
Pliki Główne Aplikacji .....	5
Src/index.js .....	5
Kod: .....	5

Style src/styles/index.scss: .....	6
Src/App.js .....	8
Kod:.....	8
Opis struktury kodu.....	9
Importowane moduły i komponenty .....	9
Funkcjonalność kodu.....	10
Komponenty.....	11
Src/components/Header/HeaderMobile/MobileHeader.js .....	11
Kod:.....	11
Style src/components/Header/HeaderMobile/MobileHeader.scss.....	16
Src/components/Header/HeaderPC/Header.js.....	22
Kod:.....	22
Style src/components/Header/HeaderPC/Header.scss: .....	25
Src/components/Footer/Footer.js.....	30
Kod:.....	30
Style src/components/Footer/Footer.scss .....	31
Komponenty Główne .....	32
Src/components/Hero/Hero.js.....	32
Kod: .....	32
Style src/components/Hero/Hero.scss: .....	33
Src/components/SectionTitle/SectionTitle.js .....	39
Kod: .....	39
Style src/components/SectionTitle/SectionTitle.scss: .....	40
src/components/Container/Container.js .....	41
Kod: .....	41
Style src/components/Container/Container.scss:.....	42
Src/components/Button/Button.js .....	43
Kod: .....	43
Style src/components/Button/Button.scss: .....	44

Src/components/ContactForm/ContactForm.js.....	46
Kod:.....	46
Style src/components/ContactForm/ContactForm.scss .....	47
Src/components/Card/Card.js.....	50
Kod:.....	50
Style src/components/Card/Card.scss:.....	51
Strony .....	53
Home src/pages/Home/Home.js.....	53
Kod:.....	54
Style src/pages/Home/Home.scss: .....	55
Komponenty:.....	56
Hero (Komponent Główny) .....	56
InfoSection {src/pages/Home/InfoSection/InfoSection.js} .....	56
Kod:.....	56
Style src/pages/Home/InfoSection/infoSection.scss:.....	58
WhyBonsaiSection {src/pages/Home/WhyBonasiSection/WhyBonsaiSection.js} ..	63
Kod:.....	64
Style src/pages/Home/WhyBonasiSection/WhyBonsaiSection.scss:.....	66
ContactSection {src/pages/Home/ContactSection/ContactSection.js} .....	68
Kod:.....	68
Style src/pages/Home/ContactSection/ContactSection.scss: .....	70
Rodzaje src/pages>TypeOfBonsai/TypeOfBonsai.js.....	72
Kod: .....	72
Style: .....	74
Komponenty:.....	75
Src/pages/TypeOfBonsai/TypesSection/TypesSection.js .....	75
Kod:.....	75
Style src/pages/TypeOfBonsai/TypesSection/TypesSection.scss: .....	77
Src/pages/TypeOfBonsai/TypesSection/SingleBonsai/SingleBonsai.js.....	78

Kod:.....	78
Style src/pages/TypeOfBonsai/TypesSection/SingleBonsai/SingleBonsai.scss:..	79
Galeria src/pages/Gallery/Gallery.js .....	84
Kod:.....	84
Style src/pages/Gallery/Gallery.scss: .....	87
Podsumowanie .....	91

## Zmienne Globalne:

Src/styles/Global.scss

### Kolory podstawowe

**\$primary-color: #40342f;**

Główny kolor tła używany w różnych sekcjach strony.

**\$primary-color-hover: #54453E;**

Kolor używany dla efektów najechania na elementy z głównym kolorem.

**\$secondary-color: #8c774f;**

Dodatkowy kolor używany dla akcentów, takich jak przyciski lub wyróżnione elementy.

**\$secondary-color-hover: #B09563;**

Kolor używany dla efektów najechania na akcenty.

### Kolory tekstu

**\$text-color: #414141;**

Kolor tekstu w treści strony.

**\$text-color-darkre: #212121;**

Kolor tekstu w nagłówkach lub elementach wymagających większego kontrastu.

## Cień

**\$shadow: 0 4px 8px rgba(0, 0, 0, 0.3);**

Używany do cieni w elementach, takich jak przyciski lub karty.

## Media Queries

**\$media-1440px: 1440px;**

Definiuje punkt przerwania dla dużych ekranów.

**\$media-1024px: 1024px;**

Definiuje punkt przerwania dla tabletów w orientacji poziomej.

**\$media-768px: 768px;**

Definiuje punkt przerwania dla tabletów w orientacji pionowej.

**\$media-425px: 425px;**

Definiuje punkt przerwania dla małych ekranów.

**\$media-375px: 375px;**

Definiuje punkt przerwania dla bardzo małych ekranów.

## Pliki Główne Aplikacji

### Src/index.js

Kod:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './styles/index.scss';
import App from './App';
import reportWebVitals from './reportWebVitals';
```

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

reportWebVitals();
```

### **Importowanie bibliotek i stylów:**

React: Główna biblioteka używana do budowy komponentów.

ReactDOM: Umożliwia renderowanie komponentów React w drzewie DOM.

index.scss: Globalne style aplikacji, zdefiniowane w pliku SCSS.

App: Główny komponent aplikacji, który zarządza całą logiką i widokiem.

reportWebVitals: Narzędzie do mierzenia wydajności aplikacji.

### **Funkcja ReactDOM.createRoot:**

Tworzy "korzeń" aplikacji w elemencie DOM o id="root". Jest to wymagane do renderowania aplikacji React w przeglądarce.

### **Renderowanie głównego komponentu App:**

React.StrictMode: Włącza dodatkowe sprawdzenia w czasie działania, aby pomóc znaleźć błędy.

### **Pomiar wydajności:**

reportWebVitals: Pozwala mierzyć wydajność i opcjonalnie logować dane, np. do analizy.

### **Style src/styles/index.scss:**

```
* , *::before, *::after {
  margin: 0;
  padding: 0;
```

```
box-sizing: inherit;  
}  
  
html {  
    box-sizing: border-box;  
    font-size: 62.5%;  
}  
  
body {  
    font-family: "Ubuntu", sans-serif;  
    color: #6D5D4B;  
    font-weight: 400;  
    font-size: 1.6rem;  
    line-height: 1.6;  
}  
  
a{  
    text-decoration: none;  
    //color: white;  
}
```

### **Resetowanie domyślnych stylów:**

\* , \*::before, \*::after: Ustawia margin i padding na 0, aby uniknąć niepożądanych odstępów w różnych przeglądarkach.

box-sizing: inherit: Ułatwia zarządzanie rozmiarami elementów, dziedzicząc box-sizing z elementu html.

### **Typografia i układ:**

#### **html:**

font-size: 62.5%: Przyjęta konwencja, która sprawia, że 1rem = 10px.

#### **body:**

Czcionka "Ubuntu", sans-serif zapewnia nowoczesny, czytelny wygląd.

Kolor tekstu #6D5D4B pasuje do stylu ziemistych odcieni.

font-size: 1.6rem i line-height: 1.6: Zapewniają odpowiednią czytelność.

## Stylizacja linków (a):

text-decoration: none; Usuwa domyślne podkreślenie, nadając linkom prosty, nowoczesny wygląd.

## Src/App.js

Kod:

```
import React, { useState, useEffect } from "react";
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './pages/Home/Home';
import TypeOfBonsai from "./pages/TypeOfBonsai/TypeOfBonsai";
import Header from "./components/Header/HeaderPC/Header";
import HeaderMobile from
"./components/Header/HeaderMobile/MobileHeader";
import Footer from "./components/Footer/Footer";
import Gallery from "./pages/Gallery/Gallery";

function App() {
  const [isMobile, setIsMobile] = useState(window.innerWidth <
1025);

  useEffect(() => {
    const handleResize = () => {
      setIsMobile(window.innerWidth < 1025);
    };

    window.addEventListener("resize", handleResize);
    return () => window.removeEventListener("resize",
handleResize);
  }, []);

  return (
    <Router>
      {isMobile ? <HeaderMobile /> : <Header />}
      <Routes>
        <Route path="/" element={<Home />} />
```

```
        <Route path="/Type-Of-Bonsai" element={<TypeOfBonsai />} />
      <Route path="/Gallery" element={<Gallery />} />
    </Routes>
    <Footer />
  </Router>
);
}

export default App;
```

### *Opis struktury kodu*

#### **Funkcja App:**

- Jest głównym komponentem aplikacji, który zarządza trasami, nagłówkiem i stanem mobilności.

#### **Struktura JSX:**

- Zawiera komponent Router, który otacza wszystkie trasy i dynamiczne nagłówki.
- Komponent Routes zarządza stronami, a każda trasa (Route) wskazuje odpowiedni komponent.

### *Importowane moduły i komponenty*

#### **React i React Hooks:**

- useState: Umożliwia przechowywanie i zarządzanie stanem komponentu.
- useEffect: Służy do wykonywania efektów ubocznych, takich jak dodawanie i usuwanie nasłuchiwanego zdarzeń.

#### **React Router:**

- Router: Zapewnia mechanizm tras w aplikacji.
- Routes: Zawiera wszystkie trasy aplikacji.
- Route: Definiuje konkretne ścieżki URL i odpowiadające im komponenty.

#### **Importowane komponenty:**

- Home: Strona główna aplikacji.

- TypeOfBonsai: Strona prezentująca różne typy bonsai.
- Header: Nagłówek dla widoków desktopowych.
- HeaderMobile: Nagłówek dla widoków mobilnych.
- Footer: Stopka wyświetlana na wszystkich stronach.
- Gallery: Strona z galerią drzewek bonsai.

### Funkcjonalność kodu

#### Stan isMobile:

- Przechowuje wartość true lub false w zależności od szerokości okna przeglądarki:
  - true: Jeśli szerokość okna jest mniejsza niż 1025px (widok mobilny).
  - false: Jeśli szerokość okna wynosi 1025px lub więcej (widok desktopowy).

#### Hook useEffect:

- Dodaje nasłuchiwanie na zdarzenie resize, aby dynamicznie zmieniać stan isMobile w zależności od szerokości okna.
- Usuwa nasłuchiwanie podczas odmontowania komponentu, aby uniknąć wycieków pamięci.

#### Dynamiczne renderowanie nagłówka:

- Wyświetla HeaderMobile w widokach mobilnych.
- Wyświetla Header w widokach desktopowych.

#### Obsługa tras:

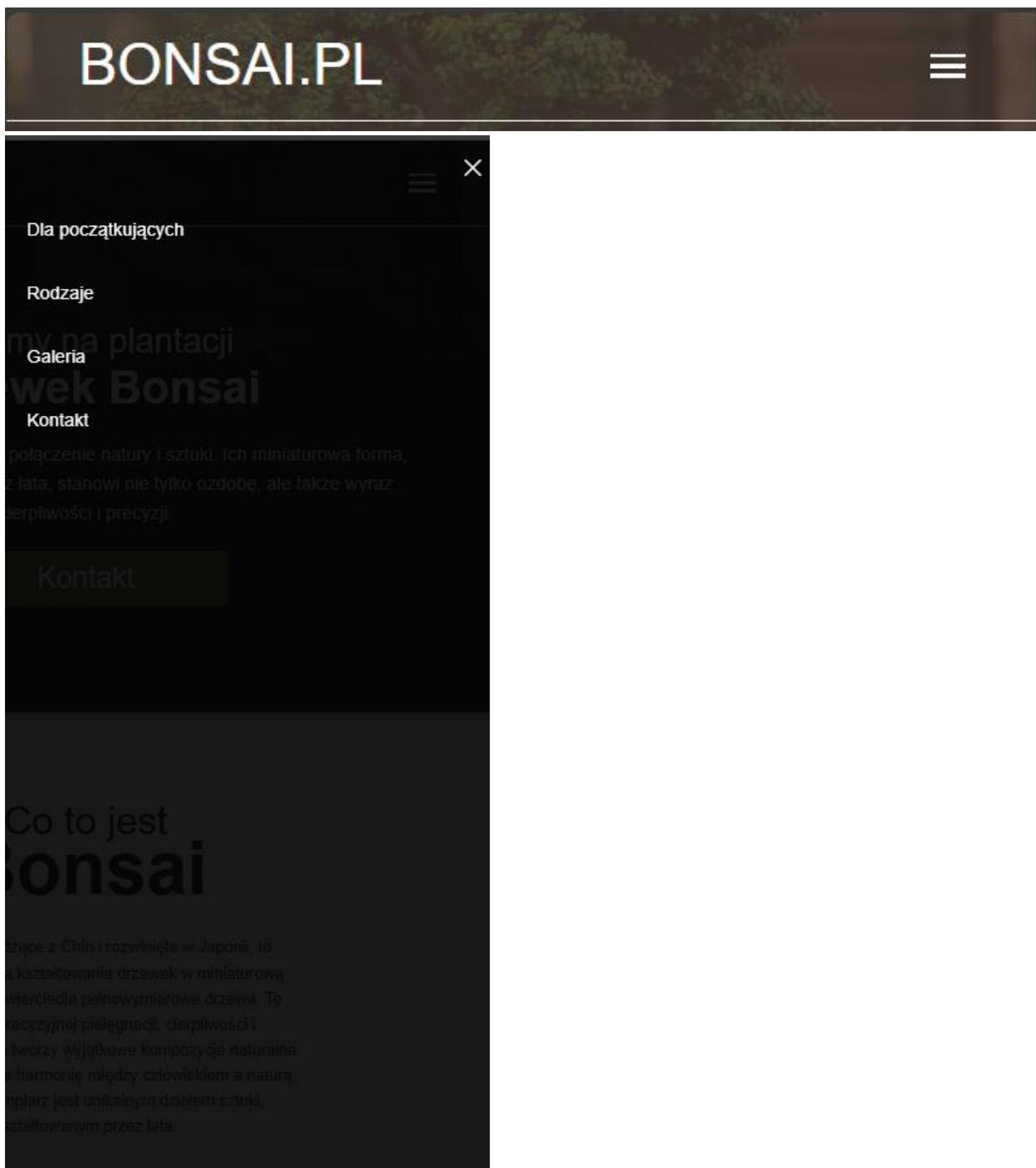
- /: Renderuje stronę główną (Home).
- /Type-Of-Bonsai: Renderuje stronę o typach bonsai (TypeOfBonsai).
- /Gallery: Renderuje stronę galerii bonsai (Gallery).

#### Stopka:

- Footer jest zawsze renderowany jako stały element na dole strony.

## Komponenty

Src/components/Header/HeaderMobile/MobileHeader.js



Kod:

```
import './MobileHeader.scss';
import { useState, useEffect } from 'react';
```

```

import { Link, useLocation, useNavigate } from "react-router-dom";
function MobileHeader() {
    const [isMenuOpen, setMenuOpen] = useState(false);
    const [isSubmenuOpen, setSubmenuOpen] = useState(false);
    const [isSticky, setIsSticky] = useState(false);
    const navigate = useNavigate();
    const location = useLocation();

    const toggleMenu = () => {
        setMenuOpen(!isMenuOpen);
        if (isMenuOpen) setSubmenuOpen(false); // Zamyka submenu, gdy
główne menu jest zamknięte
    };

    const toggleSubmenu = () => {
        setSubmenuOpen(!isSubmenuOpen);
    };

    useEffect(() => {
        const handleScroll = () => {
            if (window.scrollY > 400) {
                setIsSticky(true);
            } else {
                setIsSticky(false);
            }
        };

        window.addEventListener('scroll', handleScroll);
        return () => window.removeEventListener('scroll',
handleScroll);
    }, []);

    const scrollToSection = (id) => {
        const element = document.getElementById(id);
        console.log(id, element);
        if (element) {
            element.scrollIntoView({ behavior: 'smooth' });
            setMenuOpen(false); // Zamyka menu po przewinięciu
        } else {
    
```

```

        // Opóźnij próbę przewinięcia, jeśli element jeszcze nie
        istnieje
        setTimeout(() => {
            const delayedElement = document.getElementById(id);
            if (delayedElement) {
                delayedElement.scrollIntoView({ behavior:
                    'smooth' });
                setMenuOpen(false); // Zamiera menu po przewinięciu
            }
        }, 100);
    }

    const handleAnchorClick = (id) => {
        if (location.pathname !== "/") {
            // Jeśli nie jesteśmy na stronie głównej, nawigujemy do
            niej
            navigate("/");
            // Poczekaj, aż React Router wypisznie stronę, a potem
            przewin
            setTimeout(() => scrollToSection(id), 100);
        } else {
            scrollToSection(id);
        }
    };
}

return (
    <header className={`mobile-header ${isSticky ? 'mobile-header-
    -sticky' : ''}`}>
        <div className='mobile-header__logo'>
            <Link to="/" >Bonsai.pl</Link>
        </div>
        <div className='mobile-header__burger'
onClick={toggleMenu}>
            <span className='burger-line'></span>
            <span className='burger-line'></span>
            <span className='burger-line'></span>
        </div>
        <nav className={`mobile-header__menu ${isMenuOpen ?

```

```

'mobile-header__menu--open' : ''} }>
    <button className='mobile-header__close'
onClick={toggleMenu}>X</button>
    <ul>
        <li onClick={toggle_submenu} className={`submenu-
toggle ${is_submenuOpen ? 'submenu-toggle--open' : ''}`}>
            <span>Dla początkujących</span>
            <ul className={`submenu ${is_submenuOpen ? 'submenu--open' : ''}`}>
                <li>
                    <Link
                        to="/"
                        onClick={(e) => {
                            e.preventDefault();
                            handleAnchorClick('co-to-jest-
bonsai');
                        }}
                    >
                        Co to jest?
                    </Link>
                </li>
                <li>
                    <Link
                        to="/"
                        onClick={(e) => {
                            e.preventDefault();
                            handleAnchorClick('dlaczego-
warto-wybrac-bonasi');
                        }}
                    >
                        Dlaczego Bonsai?
                    </Link>
                </li>
            </ul>
        </li>
        <li>
            <Link to="/Type-Of-Bonsai"
onClick={toggleMenu}>
                Rodzaje

```

```

        </Link>
    </li>
    <li>
        <Link to="/Gallery" onClick={toggleMenu}>
            Galeria
        </Link>
    </li>
    <li>
        <Link
            to="/"
            onClick={(e) => {
                e.preventDefault();
                handleAnchorClick('kontakt');
            }}
        >
            Kontakt
        </Link>
    </li>
</ul>
</nav>
</header>
);
}

export default MobileHeader;

```

## Opis funkcjonalności

Stan komponentu:

- isMenuOpen: Kontroluje, czy menu główne jest otwarte.
- isSubMenuOpen: Kontroluje, czy submenu jest otwarte.
- isSticky: Informuje, czy nagłówek powinien być przyklejony podczas przewijania.

## Funkcje:

- toggleMenu: Otwiera/zamyka menu główne. Zamykając menu, zamyka też submenu.
- toggleSubMenu: Otwiera/zamyka submenu.

- handleScroll: Aktualizuje stan isSticky na podstawie pozycji przewijania.
- scrollToSection: Przewija stronę do określonej sekcji.
- handleAnchorClick: Nawiguje do strony głównej, a następnie przewija stronę do wybranej sekcji.

#### Nastąpiwanie zdarzeń:

- useEffect: Dodaje i usuwa nastąpiwanie na zdarzenie scroll, aby zaktualizować stan isSticky.

#### Renderowanie JSX:

- header: Kontener główny nagłówka z dynamicznymi klasami.
- div.mobile-header\_\_logo: Wyświetla logo strony.
- div.mobile-header\_\_burger: Ikona otwierania menu.
- nav.mobile-header\_\_menu: Menu nawigacyjne, które wyświetla listę linków i submenu.

Style src/components/Header/HeaderMobile/MobileHeader.scss

```
@import '../../../../../styles/Global';

.mobile-header {
  display: none;
  position: absolute;
  z-index: 9999;
  width: 100%;
  height: 7.5rem;
  background-color: transparent;
  border-bottom: 1px solid rgba(255, 255, 255, 0.85);
  padding: 1rem 5rem;
  color: white;
  transition: background-color 0.3s ease, top 0.3s ease;
  @media screen and (max-width: $media-425px) {
    padding: 1rem 2rem;
  }
  a{color: white}
  &--sticky {
    position: fixed;
    top: 0;
    background-color: rgba($primary-color, .95);
    animation: slideDown 0.3s ease;
  }
}
```

```
}

&__logo {
  font-size: 4rem;
  line-height: 4.5rem;
  font-weight: 400;
  display: flex;
  align-items: center;
  text-transform: uppercase;
}

&__burger {
  cursor: pointer;
  display: flex;
  flex-direction: column;
  gap: 0.4rem;

  .burger-line {
    width: 2.5rem;
    height: 0.3rem;
    background-color: white;
  }
}

&__menu {
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  position: fixed;
  top: 0;
  right: 0;
  height: 100vh;
  width: 60%;
  background-color: rgba(0, 0, 0, 0.9);
  transform: translateX(100%);
  transition: transform 0.3s ease;
  padding-top: 5rem;
  z-index: 1000;
```

```
&--open {
  transform: translateX(0);
}

ul {
  list-style: none;
  padding: 0;
  width: 100%;

li {
  padding: 1.5rem 2rem;
  color: white;
  cursor: pointer;

  &:hover {
    background-color: rgba(255, 255, 255, 0.1);
  }

  .submenu {
    display: none;
    flex-direction: column;
    padding-left: 1rem;
    padding-top: 1rem;

    li {
      padding: 0.5rem 0;
    }
  }
}

&.submenu-toggle--open .submenu {
  display: flex;
}

&__close {
  position: absolute;
  top: 1rem;
```

```

        right: 1rem;
        background: none;
        border: none;
        font-size: 2rem;
        color: white;
        cursor: pointer;
    }

    @media (max-width: $media-1024px) {
        display: flex;
        align-items: center;
        justify-content: space-between;
    }
}

@keyframes slideDown {
    from {
        top: -7.5rem;
    }
    to {
        top: 0;
    }
}

```

## Opis stylów

### Nagłówek mobilny:

- mobile-header: Główne style nagłówka.
- &--sticky: Styl przyklejonego nagłówka.

### Menu i burger:

- mobile-header\_\_burger: Styl ikony hamburgera.
- mobile-header\_\_menu: Układ i animacja menu.

### Submenu:

submenu-toggle: Kontener dla submenu, które rozwija się przy kliknięciu.

## **Menu główne (mobile-header\_\_menu)**

### **mobile-header\_\_menu:**

- Definiuje menu jako kolumnę (flex-direction: column) i ustawia je na całą wysokość widoku (height: 100vh).
- Menu jest ukrywane poza ekranem z pomocą transform: translateX(100%) i pojawia się po otwarciu.

### **mobile-header\_\_menu--open:**

- Otwiera menu poprzez zmianę transformacji na transform: translateX(0).

### **ul w menu:**

Ustawienia:

- list-style: none: Usuwa domyślne style listy.
- padding: 0: Zeruje wewnętrzne odstępy.

### **li w menu:**

Elementy listy (li) mają:

- Odstępy wewnętrzne (padding: 1.5rem 2rem).
- Kolor tekstu biały (color: white).
- Efekt najechania: background-color: rgba(255, 255, 255, 0.1).

## **Submenu (submenu)**

### **submenu:**

- Ukryte domyślnie z pomocą display: none.
- Elementy są wyświetlane jako kolumny, gdy klasa submenu-toggle--open jest aktywna.

### **submenu-toggle--open:**

- Aktywuje wyświetlanie submenu (display: flex).

### **Elementy submenu:**

- Każdy element (li) w submenu ma mniejsze odstępy (padding: 0.5rem 0).

## **Logo (mobile-header\_\_logo)**

### **mobile-header\_\_logo:**

Definiuje wygląd logotypu:

- Rozmiar tekstu: font-size: 4rem.
- Wyrównanie do środka: align-items: center.
- Wielkie litery: text-transform: uppercase.

## **Przyciski zamykania (mobile-header\_\_close)**

### **mobile-header\_\_close:**

- Pozycja absolutna w prawym górnym rogu (top: 1rem; right: 1rem).
- Duży rozmiar czcionki (font-size: 2rem) i biały kolor tekstu (color: white).

## **Animacje**

### **slideDown:**

Animacja przesuwająca nagłówek z góry na dół:

Początek: top: -7.5rem.

Koniec: top: 0.

## **Responsywność**

1. Nagłówek dla małych ekranów (@media (max-width: \$media-1024px)):
  - Wyświetla nagłówek w układzie poziomym (display: flex).
  - Ustawia elementy w rzędzie z wyrównaniem (align-items: center; justify-content: space-between).
2. Mniejsze odstępy dla bardzo małych ekranów (@media screen and (max-width: \$media-425px)):
  - Redukuje odstępy wewnętrzne do padding: 1rem 2rem.



Kod:

```
import './Header.scss';
import { useEffect, useState } from 'react';
import { Link, useLocation, useNavigate } from "react-router-dom";
function Header() {
    const [isSticky, setIsSticky] = useState(false);
    const navigate = useNavigate();
    const location = useLocation();

    useEffect(() => {
        const handleScroll = () => {
            if (window.scrollY > 400) {
                setIsSticky(true);
            } else {
                setIsSticky(false);
            }
        };
        window.addEventListener('scroll', handleScroll);
        return () => window.removeEventListener('scroll',
        handleScroll);
    }, []);

    const scrollToSection = (id) => {
        const element = document.getElementById(id);
        console.log(id,element);
        if (element) {
            element.scrollIntoView({ behavior: 'smooth' });
        } else {
            // Opóźnij próbę przewinięcia, jeśli element jeszcze nie
            istnieje
            setTimeout(() => {
                const delayedElement = document.getElementById(id);
                if (delayedElement) {
                    delayedElement.scrollIntoView({ behavior:
```

```

'smooth' });
        }
    }, 100);
}
};

const handleAnchorClick = (id) => {
    if (location.pathname !== "/") {
        // Jeśli nie jesteśmy na stronie głównej, nawigujemy do
        // niej
        navigate("/");
        // Poczekaj, aż React Router wypisznie stronę, a potem
        // przewin
        setTimeout(() => scrollToSection(id), 100);
    } else {
        scrollToSection(id);
    }
};

return (
    <header className={`header header__container ${isSticky ?
        'header--sticky' : ''}`}>
        <div className='header__logo'>
            <Link to="/" >Bonsai.pl</Link>
        </div>
        <ul className='header__menu'>
            <li className='header__menu__submenu'>
                <span>
                    Dla początkujących <span
                    className='arrow'>▼</span>
                </span>
                <ul className='submenu'>
                    <li>
                        <Link
                            to="/"
                            onClick={(e) => {
                                e.preventDefault(); // Zapobiega
                                domyльнemu działaniu linku
                                handleAnchorClick('co-to-jest-'

```

```

bonsai');
        }
    >
        Co to jest?
    </Link>
</li>
<li>
    <Link
        to="/"
        onClick={(e) => {
            e.preventDefault();
            handleAnchorClick('dlaczego-warto-
wybrac-bonasi');
        }}
    >
        Dlaczego warto wybrać bonsai?
    </Link>
</li>
</ul>
</li>
<li>
    <Link to="/Type-Of-Bonsai">Rodzaje</Link>
</li>
<li><Link to="/Gallery">Galeria</Link></li>
<li>
    <Link
        to="/"
        onClick={(e) => {
            e.preventDefault();
            handleAnchorClick('kontakt');
        }}
    >
        Kontakt
    </Link>
</li>
</ul>
</header>
);
}

```

```
export default Header;
```

## Opis funkcjonalności

### Stan komponentu:

- isSticky: Informuje, czy nagłówek powinien być przyklejony podczas przewijania.

### Funkcje:

- handleScroll: Monitoruje pozycję przewijania i ustawia isSticky na true, jeśli użytkownik przewinie stronę o więcej niż 400px.
- scrollToSection: Przewija stronę do wybranej sekcji, identyfikowanej przez id.
- handleAnchorClick: Nawiguje do strony głównej, a następnie przewija stronę do wybranej sekcji.

### Nasłuchiwanie zdarzeń:

- useEffect: Dodaje i usuwa nasłuchiwanie na zdarzenie scroll dla dynamicznej obsługi przyklejonego nagłówka.

### Renderowanie JSX:

- header: Kontener nagłówka z dynamiczną klasą header--sticky.
- div.header\_\_logo: Wyświetla logo strony jako odnośnik do strony głównej.
- ul.header\_\_menu: Menu główne z elementami i submenu.

Style src/components/Header/HeaderPC/Header.scss:

```
@import '../.../styles/Global';  
.arrow {  
  margin-left: 0.5rem;  
  font-size: 1.5rem;  
}  
  
.header {  
  box-sizing: inherit;  
  z-index: 999;  
  position: absolute;  
  top: 0;  
  width: 100%;
```

```
height: 7.5rem;
padding: 1rem 0;
color: white;
background-color: transparent;
border-bottom: 1px solid rgba(255, 255, 255, 0.85);
transition: background-color 0.3s ease, top 0.3s ease;

a{color: white}
&--sticky {
  position: fixed;
  top: 0;
  background-color: rgba($primary-color, 0.95); // Zmiana tła po
przyklejeniu
  animation: slideDown 0.3s ease; // Animacja wysuwania
}

&__container {
  display: none;
  padding: 0 5rem;
  grid-column-gap: 2rem;
}

&__logo {
  font-size: 4rem;
  line-height: 4.5rem;
  font-weight: 400;
  display: flex;
  align-items: center;
  text-transform: uppercase;
}

&__menu {
  list-style: none;
  font-size: 2rem;
  font-weight: 300;
  display: flex;
  align-items: center;
  justify-content: right;
```

```
& > * {
  padding-right: 3rem;
  text-transform: uppercase;
}

&__submenu {
  position: relative;

  .submenu {
    display: none;
    position: absolute;
    top: 100%;
    left: 0;
    background: rgba(255, 255, 255, 0.65);
    box-shadow: 0 4px 30px rgba(0, 0, 0, 0.1);
    backdrop-filter: blur(6px);
    -webkit-backdrop-filter: blur(8.2px);
    font-size: 1.6rem;
    padding: 1rem 0;
    list-style: none;
    margin: 0;
    min-width: 26rem;
  }

  li {
    & > a{color: $text-color-darkre !important;}
    padding: 1rem 2rem;
    white-space: nowrap;

    cursor: pointer;
    transition: background-color 0.5s;

    &:not(:last-child) {
      border-bottom: 1px solid $primary-color;
    }

    &:hover {
      background-color: rgba(255, 255, 255, 0.1);
    }
  }
}
```

```

        }
    
```

```

        &:hover .submenu {
            display: block;
        }
    }
}

@media (min-width: calc($media-1024px + 1px)) {
    display: grid;
    grid-template-columns: 1fr 1fr;
    @media (max-width: $media-1440px) {
        grid-template-columns: min-content 1fr;
    }
}
}

@keyframes slideDown {
    from {
        top: -7.5rem;
    }
    to {
        top: 0;
    }
}

```

## Nagłówek (.header)

- box-sizing: inherit: Dziedziczy model pudełkowy ustawiony globalnie.
- position: absolute: Ustawia nagłówek na górze strony.
- top: 0: Pozycjonuje nagłówek w górnej części widoku.
- z-index: 999: Zapewnia, że nagłówek znajduje się ponad innymi elementami strony.
- border-bottom: Dodaje subtelną linię oddzielającą nagłówek od reszty treści.
- transition: Płynne zmiany tła i położenia przy przewijaniu strony.

## Przyklejony nagłówek (.header--sticky)

- position: fixed: Utrzymuje nagłówek na stałej pozycji podczas przewijania.

- background-color: Ustawia lekko przezroczyste tło z wykorzystaniem zmiennej \$primary-color.
- animation: slideDown: Animuje nagłówek w dół podczas przyklejenia.

### **Kontener nagłówka (.header\_\_container)**

- display: none: Ukrywa kontener w mniejszych rozdzielczościach.
- padding: Dodaje przestrzeń wewnętrzną między elementami kontenera.
- grid-column-gap: Ustawia odstępy między kolumnami w układzie siatki.

### **Logo (.header\_\_logo)**

- font-size: 4rem: Definiuje duży rozmiar czcionki dla logotypu.
- line-height: 4.5rem: Dopasowuje wysokość wiersza do rozmiaru czcionki.
- text-transform: uppercase: Wymusza wielkie litery w nazwie logotypu.

### **Menu główne (.header\_\_menu)**

- list-style: none: Usuwa domyślne style listy punktowanej.
- display: flex: Ustawia elementy menu w jednej linii.
- justify-content: right: Wyrównuje elementy menu do prawej strony.
- text-transform: uppercase: Wymusza wielkie litery dla pozycji menu.

### **Pozycje menu głównego (.header\_\_menu > \*)**

- padding-right: 3rem: Dodaje odstęp między kolejnymi elementami menu.

### **Submenu (.header\_\_menu\_submenu)**

position: relative: Pozwala na umieszczenie submenu względem pozycji elementu nadzawanego.

#### **.submenu:**

- display: none: Submenu jest domyślnie ukryte.
- position: absolute: Ustawia submenu pod pozycją nadzawaną.
- background: Dodaje półprzezroczyste tło dla submenu.
- box-shadow: Tworzy efekt cienia dla submenu.
- backdrop-filter: Dodaje efekt rozmycia za submenu.
- padding: Ustawia wewnętrzne odstępy dla pozycji w submenu.

### **Elementy submenu (.submenu li)**

- padding: 1rem 2rem: Dodaje przestrzeń wewnętrzną dla pozycji submenu.

- border-bottom: Dodaje linie oddzielające kolejne elementy submenu.
- cursor: pointer: Wskazuje, że elementy są interaktywne.
- transition: background-color 0.5s: Umożliwia płynne zmiany koloru tła podczas najechania.

### Interakcja submenu (:hover)

- .submenu: Submenu staje się widoczne podczas najechania na pozycję nadzędną.
- li:hover: Zmienia kolor tła pozycji submenu podczas najechania.

### Responsywność (@media)

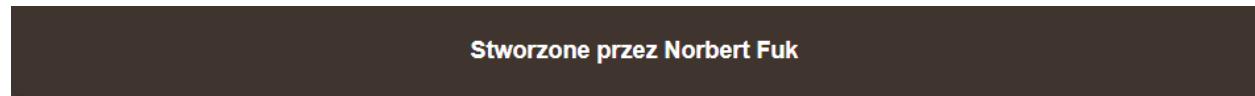
- Większe ekranы (min-width: 1025px):
- display: grid: Zmienia układ nagłówka na siatkę.
- grid-template-columns: Dzieli nagłówek na dwie kolumny (logo + menu).
- Mniejsze ekranы (max-width: 1440px):
- Zmniejsza rozmiar jednej kolumny w układzie siatki.

### Animacje

#### slideDown:

- Animuje nagłówek, przesuwając go w dół przy zmianie pozycji na sticky.

*Src/components/Footer/Footer.js*



Stworzone przez Norbert Fuk

Kod:

```
import './Footer.scss'

const Footer = () => {
  return (
    <footer className="footer">
      <p className="footer-text">
        Stworzone przez Norbert Fuk
      </p>
    </footer>
  );
}
```

```
}
```

```
export default Footer;
```

### Główna struktura:

- footer: Główny element stopki zawierający tekst z informacją o autorze.
- p.footer-text: Paragraf z treścią stopki.

### Stylizacja:

- Stylizacja znajduje się w pliku Footer.scss, który definiuje wygląd całego komponentu.

Style src/components/Footer/Footer.scss

```
@import "../../styles/Global";  
.footer{  
    width: 100%;  
    height: 7.5rem;  
    font-size: 2rem;  
    font-weight: 600;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    color: white;  
    background-color: $primary-color;  
}
```

### Stopka (.footer)

- width: 100%: Rozciąga stopkę na całą szerokość strony.
- height: 7.5rem: Ustawia wysokość stopki na stałą wartość.
- font-size: 2rem: Ustawia wielkość czcionki w stopce.
- font-weight: 600: Stosuje pogrubienie tekstu dla wyróżnienia.
- display: flex: Umożliwia elastyczne wyrównanie zawartości w stopce.
- justify-content: center: Wyrównuje zawartość stopki w poziomie do środka.

- align-items: center: Wyrównuje zawartość stopki w pionie do środka.
- color: white: Ustawia kolor tekstu na biały.
- background-color: \$primary-color: Ustawia tło stopki na główny kolor zdefiniowany w zmiennej \$primary-color z pliku Global.scss.

## Komponenty Główne

Src/components/Hero/Hero.js



Kod:

```
import './Hero.scss'
import Button from '../Button/Button'
import Container from '../Container/Container'

const Hero = ({titlePart1, titlePart2, description, buttonText,
isAnchor, href}) => {
  return(
    <section className="hero">
      <div className="hero_leftSide">
        <Container className="hero_container">
          <h1>
```

```

        <span className="hero__title hero__title--1">{titlePart1}</span>
            <br/>
        <span className="hero__title hero__title--2">{titlePart2}</span>
            </h1>
            {description && <p
className="hero__description">{description}</p>}
            {buttonText && <Button buttonText={buttonText}
href={href} isAnchor={isAnchor} />}
        </Container>
    </div>
    <div className="hero__rightSide"></div>
</section>
);
}

export default Hero;

```

Style src/components/Hero/Hero.scss:

```

@import '../styles/Global';
.hero{
    $heroTopPadding: 7.5rem;

    height: calc(50rem + $heroTopPadding);
    position: relative;
    display: grid;
    grid-template-columns: 70% 30%;
    color: white;
    box-shadow: $shadow;

    @media screen and (max-width: $media-1024px){
        grid-template-columns: 55% 45%;
    }
    @media screen and (max-width: $media-768px){
        grid-template-columns: 1fr;
        height: 50rem;
    }
}

```

```

& > * {padding-top: 7.5rem;}
&__leftSide{
    background-color: $primary-color;
    @media screen and (max-width: $media-1024px){
        //position: relative;
        background: $primary-color;
    }
    @media screen and (max-width: $media-768px){
        background:
            linear-gradient(to bottom,rgba($primary-color, 0.85) 0%,
rgba($primary-color, .99) 60%), /* Gradient na górze */
            url('../assets/heroImg945x800.webp');
        background-size: cover;
        background-position: center;
    }
}
&__rightSide{
    background:
        linear-gradient(to right,rgba(0, 0, 0, 0.5), rgba(0, 0,
0, .75)), /* Gradient na górze */
        url('../assets/heroImg945x800.webp');
    background-size: cover;
    @media screen and (max-width: $media-768px){
        display: none;
    }
}

&__container{
    box-sizing: border-box;
    position: absolute;
    left: 50%;
    transform: translateX(-50%);
    margin-left: 5rem;
    padding: 8rem 0;
    display: grid;
    grid-template-rows: max-content min-content min-content;
    gap: 2rem;
    transition: 0.2s;
}

```

```
@media screen and (max-width: $media-1440px){
  width: 90%;
}
@media screen and (max-width: $media-1024px){
  position: relative;
}
@media screen and (max-width: $media-768px){
  width: 100%;
  left: 0;
  transform: translateX(0);
  text-align: center;
  margin-left: 0;
  justify-content: center;
  justify-items: center;
}
}

&__title{
  line-height: 7rem;
  transition: 0.2s;

  @media screen and (max-width: $media-1024px){
    line-height: 6rem;
  }
  @media screen and (max-width: $media-768px){
    line-height: 4rem;
  }

  &--1{
    font-size: 6rem;
    font-weight: 300;
    @media screen and (max-width: $media-1024px){
      font-size: 5.5rem;
    }
    @media screen and (max-width: $media-768px){
      font-size: 3rem;
    }
  }
}
```

```
&--2{
    font-size: 8.5rem;
    font-weight: 600;
    @media screen and (max-width: $media-1024px){
        font-size: 6.5rem;
    }
    @media screen and (max-width: $media-768px){
        font-size: 4rem;
    }
}

&__description{

width: 65rem;
font-weight: 300;
@media screen and (max-width: $media-1024px) {
    max-width: 60rem;
}
@media screen and (max-width: $media-768px){
    width: 100%;
    text-align: center;
}
}

.Gallery .hero{
height: 40rem;
}

.Gallery .hero{
height: 40rem;
}
.typeOfBonsai .hero{
height: 50rem;
}
```

## Główna sekcja (.hero)

- height: Wysokość sekcji Hero zależy od dodanej wartości calc(50rem + \$heroTopPadding), co umożliwia dynamiczne dostosowanie.
- position: relative: Ustawia element w kontekście, aby umożliwić pozycjonowanie elementów wewnętrznych.
- display: grid: Stosuje układ siatki z dwiema kolumnami.
- grid-template-columns: 70% 30%: Lewa kolumna zajmuje 70% szerokości, prawa 30%.
- color: white: Ustawia kolor tekstu na biały.
- box-shadow: \$shadow: Dodaje cień wokół sekcji.

## Responsywność (@media)

### Do 1024px:

- grid-template-columns: 55% 45%: Zmniejsza proporcję lewej kolumny.

### Do 768px:

- grid-template-columns: 1fr: Zmienia układ na jednokolumnowy.
- height: 50rem: Ustawia stałą wysokość sekcji.

## Lewy panel (.hero\_\_leftSide)

- background-color: \$primary-color: Ustawia jednolite tło w głównym kolorze.

### Do 768px:

- background: Dodaje gradient i obraz tła, aby poprawić wygląd wizualny.
- background-size: cover: Dopasowuje obraz do pełnej szerokości i wysokości.
- background-position: center: Ustawia obraz w centrum panelu.

## Prawy panel (.hero\_\_rightSide)

- background: Zawiera gradient i obraz tła.
- background-size: cover: Dopasowuje obraz tła do rozmiaru panelu.

### Do 768px:

- display: none: Ukrywa prawy panel w widoku mobilnym.

### Kontener (.hero\_\_container)

- box-sizing: border-box: Ustawia model pudełkowy.
- position: absolute: Pozycjonuje kontener na środku lewej kolumny.
- left: 50%: Pozycjonuje kontener w środku kolumny.
- transform: translateX(-50%): Wyrównuje kontener do osi X.
- display: grid: Organizuje elementy w trzech wierszach (tytuł, opis, przycisk).
- gap: 2rem: Dodaje odstęp między elementami w kontenerze.

### Responsywność:

#### Do 1440px:

- Zmniejsza szerokość kontenera do 90%.

#### Do 768px:

- Zmienia szerokość na 100%.
- Ustawia wyrównanie tekstu na center.

### Tytuł (.hero\_\_title)

- line-height: Wysokość linii dostosowana do responsywnego wyglądu.

#### Część 1 (.hero\_\_title--1):

- font-size: Rozmiar czcionki wynosi 6rem na większych ekranach i zmniejsza się do 3rem w widoku mobilnym.
- font-weight: 300: Stosuje cieńską czcionkę.

#### Część 2 (.hero\_\_title--2):

- font-size: Większy rozmiar czcionki niż w części pierwszej (8.5rem na desktopie, 4rem na mniejszych ekranach).
- font-weight: 600: Stosuje pogrubioną czcionkę.

### Opis (.hero\_\_description)

- width: 65rem: Maksymalna szerokość opisu.

- font-weight: 300: Stosuje cieńską czcionkę.

### **Responsywność:**

#### **Do 1024px:**

- Zmniejsza szerokość do 60rem.

#### **Do 768px:**

- Zmienia szerokość na 100%.
- Ustawia wyrównanie tekstu na center.

### **Specyficzne dla podstron**

#### **Galeria (.Gallery .hero):**

- height: 40rem: Zmniejsza wysokość sekcji Hero na podstronie galerii.

#### **Rodzaje bonsai (.typeOfBonsai .hero):**

- height: 50rem: Ustawia stałą wysokość dla sekcji Hero na tej podstronie.

Src/components/SectionTitle/SectionTitle.js

# Co to jest Bonsai

Kod:

```
import React from 'react';
import PropTypes from 'prop-types';
import './SectionTitle.scss'

const SectionTitle = ({className = '', titlePart1, titlePart2 }) => {
  return (
    <h2 className={`section-title ${className}`}>
      <span className={'section-title--1'}>{titlePart1}</span>
      <br/>
      <span className={'section-title--2'}>{titlePart2}</span>
    </h2>
  )
}

SectionTitle.propTypes = {
  className: PropTypes.string,
  titlePart1: PropTypes.string,
  titlePart2: PropTypes.string
}

export default SectionTitle
```

```
    );
};

SectionTitle.propTypes = {
  className: PropTypes.string
};

export default SectionTitle;
```

### Funkcja komponentu:

SectionTitle: Komponent wyświetlający dwuczęściowy tytuł sekcji z możliwością nadania dodatkowych klas.

### Propsy:

- className: Opcjonalna klasa CSS, pozwalająca na modyfikację stylów zewnętrznych.
- titlePart1: Tekst pierwszej części tytułu.
- titlePart2: Tekst drugiej części tytułu.

### Struktura JSX:

- h2.section-title: Główne opakowanie tytułu z dynamicznie dodaną klasą CSS.
- span.section-title--1: Pierwsza część tytułu.
- span.section-title--2: Druga część tytułu.

### PropTypes:

- className: Weryfikowany jako string za pomocą PropTypes.

### Style src/components/SectionTitle/SectionTitle.scss:

```
@import "../../styles/Global";

.section-title{
  color: $text-color;
  line-height: 3.8rem;
```

```
&--1{
  font-weight: 400;
  font-size: 3.4rem;
}
&--2{
  font-size: 6.4rem;
}
}
```

### Główna klasa (.section-title)

- color: \$text-color: Ustawia kolor tekstu na zmienną \$text-color zdefiniowaną w stylach globalnych.
- line-height: 3.8rem: Dostosowuje wysokość linii dla lepszej czytelności.

### Pierwsza część tytułu (.section-title--1)

- font-weight: 400: Stosuje standardową wagę czcionki.
- font-size: 3.4rem: Ustawia średni rozmiar czcionki dla pierwszej części tytułu.

### Druga część tytułu (.section-title--2)

- font-size: 6.4rem: Ustawia większy rozmiar czcionki, aby wyróżnić drugą część tytułu.

src/components/Container/Container.js

Kod:

```
import React from 'react';
import PropTypes from 'prop-types';
import './Container.scss'

const Container = ({ children, className = '' }) => {
  return (
    <div className={`container ${className}`}>
      {children}
    </div>
  )
}
```

```
    );
};

Container.propTypes = {
  children: PropTypes.node,
  className: PropTypes.string
};

export default Container;
```

### Funkcja komponentu:

Container: Uniwersalny komponent opakowujący, używany do zarządzania układem i szerokością treści na stronie.

### Propsy:

- children: Elementy podrzędne przekazane do kontenera.
- className: Opcjonalna klasa CSS, pozwalająca na dodatkową modyfikację stylów.

### Struktura JSX:

- div.container: Główny element kontenera, w którym renderowane są dzieci (children).

### PropTypes:

- children: Weryfikowany jako dowolny węzeł React za pomocą PropTypes.node.
- className: Weryfikowany jako string za pomocą PropTypes.string.

### Style src/components/Container/Container.scss:

```
@import '..../styles/Global';
.container{
  width: 120rem;
  @media screen and (max-width: $media-1440px){
    width: 90%;
  }
  margin: 0 auto;
}
```

## Główna klasa (.container)

- width: 120rem: Ustawia maksymalną szerokość kontenera na 120rem.
- margin: 0 auto: Centruje kontener w poziomie, ustawiając równe marginesy z obu stron.

## Responsywność (@media)

### Do 1440px:

- width: 90%: Zmniejsza szerokość kontenera, aby lepiej pasowała do mniejszych ekranów.

Src/components/Button/Button.js

Zobacz rodzaje

Kod:

```
import './Button.scss'
import {Link} from "react-router-dom";

const Button = ({isAnchor, href, buttonText}) => {
  if (isAnchor) {
    return (
      <button className="button" onClick={() =>
window.location.hash = href}>
        {buttonText}
      </button>
    );
  }
}
```

```
        return (
          <Link className="button" to={href}>{buttonText}</Link>
        );
      }

export default Button;
```

Funkcja komponentu:

Button: Komponent przycisku, który działa jako odnośnik (`<Link>`) lub kotwica (`<button>`), w zależności od wartości prop `isAnchor`.

#### Propsy:

- `isAnchor`: Boolean, określa, czy komponent powinien działać jako kotwica na stronie (np. do przewijania).
- `href`: Adres URL lub identyfikator sekcji, do której ma prowadzić.
- `buttonText`: Tekst wyświetlany na przycisku.

#### Struktura JSX:

- `button.button`: Renderowany jako element HTML `<button>`, gdy `isAnchor` jest `true`.
- `Link.button`: Renderowany jako odnośnik React Router (`<Link>`), gdy `isAnchor` jest `false`.

Style src/components/Button/Button.scss:

```
@import '../styles/Global';

.button{
  color: white;
  width: 25rem;
  padding: 1rem 0.5rem;
  border-radius: 0.5rem;
  border: none;
  font-size: 2.5rem;
  background-color: $secondary-color;
  text-decoration: none;
  text-align: center;

  &:first-letter {text-transform: uppercase}
```

```
&:visited{  
    color: white;  
}  
&:hover{  
    background-color: $secondary-color-hover;  
}  
}
```

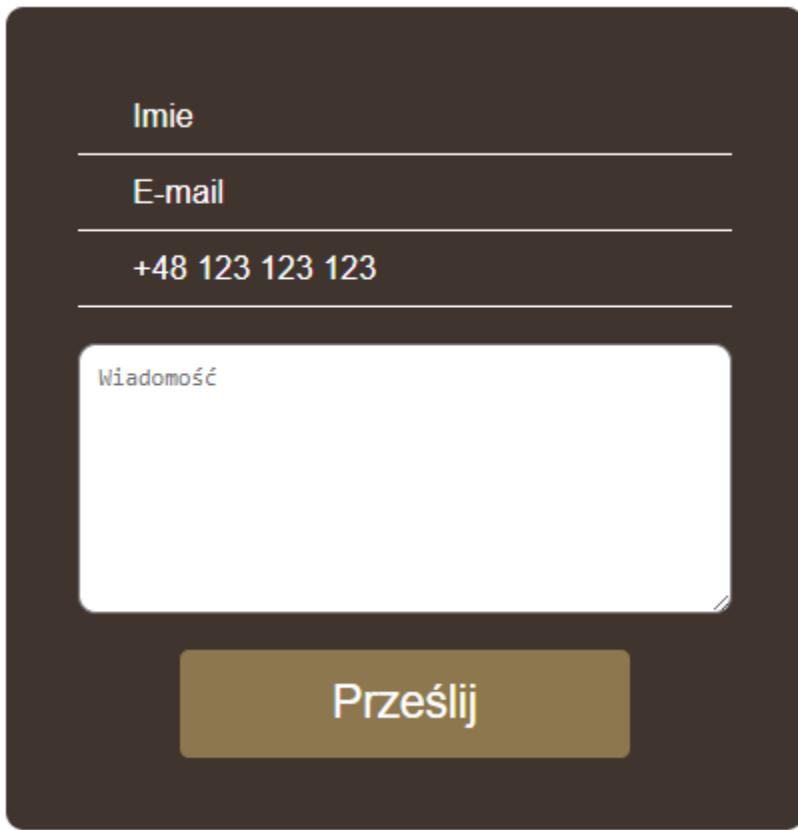
### Główna klasa (.button)

- color: white: Tekst w przycisku jest biały.
- width: 25rem: Stała szerokość przycisku.
- padding: 1rem 0.5rem: Dodaje wewnętrzne odstępy w pionie i poziomie.
- border-radius: 0.5rem: Zaokrąglą rogi przycisku.
- border: none: Usuwa domyślną ramkę przycisku.
- font-size: 2.5rem: Ustawia rozmiar tekstu w przycisku.
- background-color: \$secondary-color: Ustawia kolor tła na wartość zmiennej \$secondary-color.
- text-decoration: none: Usuwa podkreślenie tekstu (dotyczy linków).
- text-align: center: Wyrównuje tekst w środku przycisku.

### Interakcje (:first-letter, :visited, :hover)

- :first-letter {text-transform: uppercase}: Automatycznie zamienia pierwszą literę tekstu na wielką literę.
- :visited: Zachowuje biały kolor tekstu, nawet po kliknięciu.
- :hover: Zmienia kolor tła na wartość zmiennej \$secondary-color-hover.

## Src/components/ContactForm/ContactForm.js



Kod:

```
import './ContactForm.scss'
import Button from '../Button/Button';

const ContactForm = () => {
  return (
    <form className="contact-form">
      <div className="contact-form__container">
        <input className="contact-form__input" type="text"
name="name" placeholder="Imię"/>
        <input className="contact-form__input" type="text"
name="email" placeholder="E-mail"/>
        <input className="contact-form__input" type="number"
name="telephoneNumber" placeholder="+48 123 123 123"/>
      </div>
      <textarea className="contact-form__area" name="message"
placeholder="Wiadomość"/>
```

```

        <Button className="contact-form__button"
buttonText="Prześlij" type="submit">Submit</Button>
    </form>
)
};

export default ContactForm;

```

### Funkcja komponentu:

ContactForm: Formularz kontaktowy umożliwiający użytkownikom przesyłanie wiadomości z informacjami takimi jak imię, e-mail, numer telefonu oraz wiadomość.

### Struktura formularza:

- form.contact-form: Główny kontener formularza.
- div.contact-form\_\_container: Kontener dla pól tekstowych (imię, e-mail, numer telefonu).
- textarea.contact-form\_\_area: Pole do wpisywania wiadomości.
- Button.contact-form\_\_button: Przycisk do przesyłania formularza

### Style src/components/ContactForm/ContactForm.scss

```

@import "../../styles/Global";
.contact-form {
  padding: 4rem;
  background: $primary-color;
  border-radius: 1rem;
  display: grid;
  grid-template-rows: max-content 1fr max-content;
  row-gap: 1rem;
  margin: 0 4rem;
  justify-items: center;
  @media screen and (max-width: $media-1024px) {
    max-width: 40rem;
  }

  input, textarea{width: 100%}

  &__input {

```

```

padding: 1rem 3rem;
background: transparent;
border-bottom: 2px solid white !important;
border: none;
font-family: inherit;
font-size: 1.8rem;
font-weight: 300 !important;
&::placeholder {color:white;}
}
&__container{

}

&__area{
border-radius: inherit;
min-height: 15rem;
margin: 1rem 0;
padding: 1rem;
}
&__button {
justify-self: center;
}
}

```

### Główna klasa (.contact-form)

- padding: 4rem: Dodaje odstępy wewnętrzne wokół formularza.
- background: \$primary-color: Ustawia kolor tła na wartość zmiennej \$primary-color.
- border-radius: 1rem: Zaokrąglą rogi formularza.
- display: grid: Organizuje elementy formularza w siatce.
- grid-template-rows: max-content 1fr max-content: Ustawia siatkę w trzech wierszach: pola, obszar wiadomości i przycisk.
- row-gap: 1rem: Dodaje odstępy między wierszami.
- margin: 0 4rem: Dodaje marginesy poziome.
- justify-items: center: Wyrównuje elementy w poziomie.

### Responsywność:

**Do 1024px:**

- max-width: 40rem: Ogranicza szerokość formularza.

### **Pola tekstowe i obszar wiadomości (input, textarea)**

- width: 100%: Ustawia szerokość pól na całą szerokość kontenera.

### **Pola wejściowe (.contact-form\_\_input)**

- padding: 1rem 3rem: Dodaje przestrzeń wewnętrzną w polach.
- background: transparent: Ustawia przezroczyste tło.
- border-bottom: 2px solid white: Ustawia dolną ramkę na biało.
- border: none: Usuwa pozostałe domyślne ramki.
- font-family: inherit: Dziedziczy styl czcionki z nadrzędnego elementu.
- font-size: 1.8rem: Ustawia wielkość czcionki.
- font-weight: 300: Ustawia cieśszą wagę czcionki.

### **Placeholder:**

- color: white: Placeholder są wyświetlane w białym kolorze.

### **Obszar wiadomości (.contact-form\_\_area)**

- border-radius: inherit: Dziedziczy zaokrąglenie od nadrzędnego elementu.
- min-height: 15rem: Ustawia minimalną wysokość pola tekstowego.
- margin: 1rem 0: Dodaje odstępy nad i pod elementem.
- padding: 1rem: Dodaje przestrzeń wewnętrzną.

### **Przycisk (.contact-form\_\_button)**

- justify-self: center: Wyrównuje przycisk w poziomie w środku.

## Src/components/Card/Card.js



Kod:

```
import "./Card.scss"

const Card = ({ title, description, photo }) => {
  return (
    <section className="Card">
      <img src={photo} className="Card__icon" alt={title} />
      <h3 className="Card__title">{title}</h3>
      <p className="Card__description">{description}</p>
    </section>
  );
}

export default Card;
```

### Funkcja komponentu:

Card: Reprezentuje kartę informacyjną, która zawiera tytuł, opis oraz ikonę/obrazek.

### Propsy:

- title: Tytuł karty wyświetlany w nagłówku (h3).

- `description`: Opis wyświetlany pod tytułem.
- `photo`: Ścieżka do obrazu lub ikony, która zostanie wyświetlona na karcie.

### Struktura JSX:

- `section.Card`: Główny kontener karty.
- `img.Card__icon`: Obraz/ikona wyświetlana na karcie.
- `h3.Card__title`: Nagłówek wyświetlający tytuł karty.
- `p.Card__description`: Paragraf z opisem karty.

Style src/components/Card/Card.scss:

```
@import '../styles/Global';
.whyBonsai{
  .Card{
    min-height: 30rem;
    background: white;
    border-radius: 1rem;
    padding: 2.5rem;
    text-align: center;
    font-size: 1.8rem;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.35);
    &__icon{
      width: 15rem;
      @media (max-width: $media-1024px){
        width: 12rem;
      }
    }
    &__title{
      @media (max-width: $media-1024px){
        font-size: 1.8rem;
      }
    }
    &__description{
      font-size: 1.5rem;
      @media (max-width: $media-1024px){
        font-size: 1.3rem;
      }
    }
  }
}
```

```
}
```

### Główna klasa (.Card)

- min-height: 30rem; Ustawia minimalną wysokość karty, aby zapewnić spójny wygląd.
- background: white; Ustawia białe tło karty.
- border-radius: 1rem; Zaokrąglą rogi karty.
- padding: 2.5rem; Dodaje przestrzeń wewnętrzną wewnątrz karty.
- text-align: center; Wyrównuje zawartość karty do środka.
- font-size: 1.8rem; Ustawia podstawowy rozmiar czcionki.
- box-shadow: 0 4px 6px rgba(0, 0, 0, 0.35); Dodaje subtelny cień dla efektu głębi.

### Ikona/Obraz (.Card\_\_icon)

- width: 15rem; Ustawia szerokość obrazu.

#### Responsywność:

##### Do 1024px:

- width: 12rem; Zmniejsza szerokość obrazu na mniejszych ekranach.

### Tytuł (.Card\_\_title)

#### Responsywność:

##### Do 1024px:

- font-size: 1.8rem; Zmniejsza rozmiar czcionki na mniejszych ekranach.

### Opis (.Card\_\_description)

- font-size: 1.5rem; Ustawia domyślny rozmiar tekstu opisu.

#### Responsywność:

##### Do 1024px:

- font-size: 1.3rem: Zmniejsza rozmiar tekstu na mniejszych ekranach.

## Strony

### Home src/pages/Home/Home.js



#### Co to jest Bonsai

Bonsai, pochodzące z Chin i rozwinięte w Japonii, to starożytna sztuka kształtowania drzewek w miniaturową formę, która odzwierciedla pełnowymiarowe drzewa. To połączenie precyzyjnej pielęgnacji, cierpliwości i kreatywności, które tworzy wyjątkowe kompozycje naturalne. Bonsai symbolizuje harmonię między człowiekiem a naturą, a każdy egzemplarz jest unikalnym dziełem sztuki, kształtowanym przez lata.



Zobacz rodzaje



Imię	<input type="text"/>
E-mail	<input type="text"/>
+48 123 123 123	<input type="text"/>
Adres	<input type="text"/>
<b>Przeslij</b>	

#### Masz jakieś pytania Napisz do nas!

Chcesz dowiedzieć się więcej o pielęgnacji bonsai? A może masz pytania dotyczące wyboru odpowiedniego drzewka? Skontaktuj się z nami!

E-mail: kontakt@bonsaiart.pl  
Telefon: +48 123 456 789

Jesteśmy dostępny od poniedziałku do piątku, w godzinach 9:00 - 17:00. Chętnie pomożemy Ci na każdym etapie przygody z bonsai!

Stworzone przez Norbert Fuk

Kod:

```
import './Home.scss';
import Hero from '../../components/Hero/Hero';
import InfoSection from './InfoSection/InfoSection';
import WhyBonsaiSection from './WhyBonsaiSection/WhyBonsaiSection';
import ContactSection from "./ContactSection/ContactSection";
import {useEffect} from "react";

const heroData = {
    titlePart1: "Witamy na plantacji",
    titlePart2: "Drzewek Bonsai",
    description: "Drzewka bonsai to wyjątkowe połączenie natury i sztuki. Ich miniaturowa forma, misternie kształtowana przez lata, stanowi nie tylko ozdobę, ale także wyraz cierpliwości i precyzji.",
    buttonText: "kontakt",
    isAnchor: true,
    href: "#kontakt"
};

const Home = () => {
    useEffect(() => {
        document.title = "Home - Bonsai.pl"; // Ustaw nazwę strony
    }, []);
    return (
        <main className="home">
            <Hero
                titlePart1={heroData.titlePart1}
                titlePart2={heroData.titlePart2}
                description={heroData.description}
                buttonText={heroData.buttonText}
                isAnchor={heroData.isAnchor}
                href={heroData.href}
            />
            <InfoSection id="co-to-jest-bonsai" />
            <WhyBonsaiSection id="dlaczego-warto-wybrac-bonasi" />
            <ContactSection id="kontakt" />
        </main>
    );
}
```

```
};  
  
export default Home;
```

#### Funkcja komponentu:

Home: Główna strona aplikacji, zawierająca sekcje wprowadzające i informacyjne dotyczące bonsai.

#### Komponenty używane w Home:

- Hero: Sekcja główna z tytułem, opisem i przyciskiem prowadzącym do sekcji kontaktowej.
- InfoSection: Sekcja informacyjna, opisująca, czym jest bonsai.
- WhyBonsaiSection: Sekcja wyjaśniająca, dlaczego warto wybrać bonsai.
- ContactSection: Sekcja kontaktowa umożliwiająca nawiązanie kontaktu.

#### Dane statyczne (heroData):

- titlePart1: Pierwsza część tytułu, wyświetlana w sekcji Hero.
- titlePart2: Druga część tytułu.
- description: Opis pod tytułem, wprowadzający użytkownika w tematykę bonsai.
- buttonText: Tekst na przycisku prowadzącym do sekcji kontaktowej.
- isAnchor: Określa, że przycisk Hero działa jako kotwica.
- href: Kotwica prowadząca do sekcji #kontakt.

#### Efekty (useEffect):

- Ustawia tytuł dokumentu na "Home - Bonsai.pl" przy załadowaniu strony.

#### Struktura JSX:

- main.home: Główny kontener strony Home, zawierający wszystkie sekcje.

#### Style src/pages/Home/Home.scss:

```
@import ' ../../styles/Global' ;  
  
.home{  
    width:100%;  
}
```

## Główna klasa (.home)

- width: 100%: Rozciąga główny kontener na pełną szerokość strony.

### Komponenty:

*Hero (Komponent Główny)*

[\[Opis sekcji Hero\]](#)

*InfoSection {src/pages/Home/InfoSection/InfoSection.js}*

## Co to jest **Bonsai**

Bonsai, pochodzące z Chin i rozwinięte w Japonii, to starożytna sztuka kształtuowania drzewek w miniaturową formę, która odzwierciedla pełnowymiarowe drzewa. To połączenie precyzyjnej pielęgnacji, cierpliwości i kreatywności, które tworzy wyjątkowe kompozycje naturalne. Bonsai symbolizuje harmonię między człowiekiem a naturą, a każdy egzemplarz jest unikalnym dziełem sztuki, kształtowanym przez lata.

[Zobacz rodzaje](#)



### Kod:

```
import "./infoSection.scss"
import SectionTitle from
"../../../../components/SectionTitle/SectionTitle";
import Button from "../../../../components/Button/Button";
import Container from "../../../../components/Container/Container";
import img1 from "../../../../assets/infosection-photo-1.webp";
import img2 from "../../../../assets/infosection-photo-2.webp";

const InfoSection = ({id}) =>{
```

```

    return (
      <section id={id} className="info-section">
        <Container className="info-section__container">
          <div className="info-section__content">
            <SectionTitle
              className="info-section__title"
              titlePart1="Co to jest"
              titlePart2="Bonsai"
            ></SectionTitle>
            <p className="info-section__content__description">
              Bonsai, pochodzące z Chin i rozwinięte w
            Japonii, to starożytna sztuka kształtowania drzewek w
              miniaturową formę, która odzwierciedla
            pełnowymiarowe drzewa. To połączenie precyzyjnej
              pielęgnacji, cierpliwości i kreatywności,
            które tworzy wyjątkowe kompozycje naturalne. Bonsai
              symbolizuje harmonię między człowiekiem a
            naturą, a każdy egzemplarz jest unikalnym dziełem
              sztuki, kształtowanym przez lata.
            </p>
            <Button isAnchor={false} href={"/pages"}>
              buttonText="Zobacz rodzaje"</Button>
            </div>
            <div className="info-section__photos">
              <img src={img1} className="info-section__photos--1" alt="bonsai" />
              <div className="info-section__photos__background"></div>
              <img src={img2} className="info-section__photos--2" alt="bonsai" />
            </div>
          </Container>
        </section>
    )
};

export default InfoSection;

```

### Funkcja komponentu:

InfoSection: Sekcja informacyjna przedstawiająca wprowadzenie do bonsai, zawierająca tytuł, opis, przycisk nawigacyjny oraz zdjęcia.

### Propsy:

- id: Identyfikator sekcji, umożliwiający nawigację do niej za pomocą kotwicy.

### Komponenty używane w InfoSection:

- SectionTitle: Wyświetla tytuł sekcji w dwóch częściach.
- Button: Przycisk nawigacyjny prowadzący do strony z rodzajami bonsai.
- Container: Kontener zarządzający szerokością i układem treści.

### Dodatkowe elementy:

- img1, img2: Obrazy bonsai wyświetlane w sekcji.
- info-section\_\_photos\_\_background: Element dekoracyjny z tłem w kolorze głównym.

### Struktura JSX:

- section.info-section: Główny kontener sekcji.
- div.info-section\_\_content: Kontener dla tekstu i przycisku.
- div.info-section\_\_photos: Kontener dla zdjęć i elementu tła.

Style src/pages/Home/InfoSection/infoSection.scss:

```
@import '../.../styles/Global';
.info-section{
    width: 100%;
    padding: 7.5rem 0;

    &__container{
        display: grid;
        grid-template-columns: 60% 40%;
        column-gap: 1rem;
        @media screen and (max-width: $media-768px){
            grid-template-columns: 1fr;
            grid-template-rows: max-content max-content;
            justify-items: center;
        }
    }
}
```

```
    align-items: center;
    //width: 100% !important;
}
}

&__content{
  display: grid;
  row-gap: 2.5rem;
  max-width: 65rem;
  @media screen and (max-width: $media-768px){
    grid-template-rows: min-content max-content min-content;
    justify-items: center;
    text-align: center;
    max-width: 38rem;
  }
}

&__description{
  font-size: 2rem;
  margin-right: 7rem;
  @media screen and (max-width: $media-1024px){
    font-size: 1.6rem;
  }
  @media screen and (max-width: $media-768px){
    font-size: 1.4rem;
    margin: 0;
    width: 380px;
  }
}

&__photos{
  position: relative;
  transform: scale(0.9);
```

```
@media screen and (max-width: $media-768px){
  padding: 4rem 0;
  height: 35rem;
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-gap: 3rem;
  justify-items: center;
}
& > *{
  height: 30rem;
  //width: 20rem;
  position: absolute;
  transition: .5s;
  @media screen and (max-width: $media-1024px){
    height: 25rem;
  }
  @media screen and (max-width: $media-768px){
    position: relative;
    left: 0;
    top: 0;
    height: 20rem;
  }
}
& > img {
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.35);
}

&--1{
  z-index: 20;
}
&__background{
  width: 27rem;
  height: 33rem;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%) scale(1.2) translateY(-20px);
  background: $primary-color;
  @media screen and (max-width: $media-768px){
```

```
        display: none;
    }
}
&--2{
    z-index: 30;
    right: 0;
    bottom: 0;
}

&--1, &--2{
    &:hover{
        z-index: 999;
    }
}
}
```

### Główna klasa (.info-section)

- width: 100%: Rozciąga sekcję na pełną szerokość strony.
- padding: 7.5rem 0: Dodaje przestrzeń nad i pod sekcją.

### Kontener (.info-section\_\_container)

- display: grid: Ustawia siatkę z dwiema kolumnami.
- grid-template-columns: 60% 40%: Lewa kolumna zajmuje 60%, prawa 40%.

### Responsywność:

#### Do 768px:

- Układ zmienia się na jednokolumnowy.
- Elementy są wyśrodkowane (justify-items: center; align-items: center).

### Treść (.info-section\_\_content)

- display: grid: Organizuje treść w siatce z wierszami.
- row-gap: 2.5rem: Ustawia odstępy między wierszami.

- max-width: 65rem; Ogranicza szerokość tekstu.

#### **Podklasa .info-section\_\_content\_\_description:**

- font-size: 2rem; Ustawia rozmiar tekstu opisu.
- margin-right: 7rem; Dodaje odstęp po prawej stronie.

#### **Responsywność:**

##### **Do 1024px:**

- Zmniejsza rozmiar czcionki do 1.6rem.

##### **Do 768px:**

- Dostosowuje szerokość do 38rem.
- Tekst jest wyśrodkowany.

#### **Zdjęcia (.info-section\_\_photos)**

- position: relative; Ustawia zdjęcia względem kontenera.
- transform: scale(0.9); Skalowanie zmniejszające element.

#### **Zdjęcia wewnętrzne (& > \*):**

- height: 30rem; Ustawia wysokość zdjęć.
- position: absolute; Pozycjonowanie względem kontenera zdjęć.
- box-shadow: Dodaje cień wokół zdjęć.

#### **Responsywność:**

##### **Do 1024px:**

- Wysokość zdjęć zmniejsza się do 25rem.

##### **Do 768px:**

- position: relative; Usuwa absolutne pozycjonowanie.
- Wysokość zdjęć zmniejsza się do 20rem.

#### **Zdjęcie 1 (.info-section\_\_photos--1):**

- z-index: 20; Ustawia kolejność warstw.

#### **Zdjęcie 2 (.info-section\_\_photos--2):**

- z-index: 30; Umieszcza zdjęcie nad tłem.
- right: 0; bottom: 0; Pozycjonuje zdjęcie w prawym dolnym rogu kontenera.

### Tło (.info-section\_\_photos\_\_background)

- width: 27rem; Szerokość tła dekoracyjnego.
- height: 33rem; Wysokość tła dekoracyjnego.
- background: \$primary-color; Kolor tła w kolorze głównym.
- transform: translate(-50%, -50%) scale(1.2) translateY(-20px); Pozycjonowanie i skalowanie tła.

### Responsywność:

#### Do 768px:

- Element tła jest ukrywany (display: none).

*WhyBonsaiSection {src/pages/Home/WhyBonasiSection/WhyBonsaiSection.js}*

Dlaczego  
**Warto wybrać Bonsai?**



**Redukcja stresu**  
Pielęgnacja bonsai pomaga się wyciszyć i skupić.



**Satyfakcja z tworzenia**  
Kształtowanie bonsai to twórczy proces, który daje wiele radości i poczucia dumy.



**Estetyka i harmonia**  
Pielęgnacja bonsai pomaga się wyciszyć i skupić.

Kod:

```
import './WhyBonsaiSection.scss';
import Container from "../../components/Container/Container";
import SectionTitle from
"../../components/SectionTitle/SectionTitle";
import Card from "../../components/Card/Card";
import ikona1 from "../../assets/ikona1.webp"
import ikona2 from "../../assets/ikona2.webp"
import ikona3 from "../../assets/ikona3.webp"
const CardData = {
  1: {
    title: "Redukcja stresu",
    photo: ikona1, // Podaj właściwą nazwę pliku
    description: "Pielęgnacja bonsai pomaga się wyciszyć i
skupić.",
  },
  2: {
    title: "Satysfakcja z tworzenia",
    photo: ikona2, // Podaj właściwą nazwę pliku
    description: "Kształtowanie bonsai to twórczy proces, który
daje wiele radości i poczucia dumy.",
  },
  3: {
    title: "Estetyka i harmonia",
    photo: ikona3, // Podaj właściwą nazwę pliku
    description: "Pielęgnacja bonsai pomaga się wyciszyć i
skupić.",
  },
};

const WhyBonsaiSection = ({id}) => {
  return (
    <section id={id} className="whyBonsai">
      <Container className="whyBonsai__container">
        <SectionTitle
          className="whyBonsai__title"
          titlePart1="Dlaczego"
          titlePart2="Warto wybrać Bonsai?" />
        <Container className="whyBonsai__cardContainer">
```

```

        <Card
            title={CardData[1].title}
            description={CardData[1].description}
            photo={CardData[1].photo}
        />
        <Card
            title={CardData[2].title}
            description={CardData[2].description}
            photo={CardData[2].photo}
        />
        <Card
            title={CardData[3].title}
            description={CardData[3].description}
            photo={CardData[3].photo}
        />
    </Container>
</Container>
</section>
);
};

export default WhyBonsaiSection;

```

### Funkcja komponentu:

WhyBonsaiSection: Sekcja przedstawiająca zalety wyboru bonsai z trzema kluczowymi aspektami zaprezentowanymi w formie kart.

### Propsy:

- id: Identyfikator sekcji, używany do nawigacji kotwicznej.

### Dane (CardData):

- Obiekt CardData przechowuje informacje o tytułach, opisach i obrazach dla trzech kart w sekcji.

### Komponenty używane w WhyBonsaiSection:

- Container: Opakowuje sekcję i zarządza szerokością treści.

- **SectionTitle:** Wyświetla tytuł sekcji w dwóch częściach.
- **Card:** Reprezentuje pojedynczą kartę z tytułem, opisem i obrazem.

### **Struktura JSX:**

- **section.whyBonsai:** Główny kontener sekcji.
- **Container.whyBonsai\_\_container:** Kontener zarządzający układem sekcji.
- **Container.whyBonsai\_\_cardContainer:** Kontener zarządzający układem kart.

Style src/pages/Home/WhyBonasiSection/WhyBonsaiSection.scss:

```
@import '../../styles/Global';

.whyBonsai{
  width: 100%;
  padding: 7.5rem 0;
  background:
    linear-gradient(to top, rgba(255, 255, 255, 0.8), rgba(255, 255, 255, 0.9)),
    url("../../assets/whyBonsaiSection.webp");
  background-size: cover;
  background-position: center;
  &__container{
    display: grid;
    row-gap: 6rem;
  }

  &__title{
    text-align: center;
  }

  &__cardContainer{
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    column-gap: 5rem;
    @media screen and (max-width: $media-1024px){
      column-gap: 2rem;
    }

    @media screen and (max-width: $media-768px){
```

```
    max-width: 40rem;
    row-gap: 2rem;
    grid-template-columns: 1fr;
}
}
}
```

### Główna klasa (.whyBonsai)

- width: 100%: Rozciąga sekcję na pełną szerokość strony.
- padding: 7.5rem 0: Dodaje przestrzeń nad i pod sekcją.
- background: Ustawia gradient i obraz jako tło.
  - Gradient: Dodaje subtelne rozjaśnienie od góry.
  - Obraz: Wyświetla obraz w tle, skalowany do pełnej szerokości (background-size: cover) i centrowany (background-position: center).

### Kontener główny (.whyBonsai\_\_container)

- display: grid: Ustawia elementy w układzie siatki.
- row-gap: 6rem: Dodaje odstępy między wierszami.

### Tytuł sekcji (.whyBonsai\_\_title)

- text-align: center: Wyśrodkowuje tekst w poziomie.

### Kontener kart (.whyBonsai\_\_cardContainer)

- display: grid: Ustawia karty w układzie siatki.
- grid-template-columns: repeat(3, 1fr): Tworzy trzy kolumny o równych proporcjach.
- column-gap: 5rem: Dodaje odstępy między kolumnami.

### Responsywność:

#### Do 1024px:

- Zmniejsza odstępy między kolumnami do 2rem.

#### Do 768px:

- max-width: 40rem; Ogranicza szerokość kontenera.
- row-gap: 2rem; Zmniejsza odstępy między wierszami.
- grid-template-columns: 1fr; Zmienia układ na jednokolumnowy.

ContactSection {src/pages/Home/ContactSection/ContactSection.js}

## Masz jakieś pytania **Napisz do nas!**

Chcesz dowiedzieć się więcej o pielęgnacji bonsai? A może masz pytania dotyczące wyboru odpowiedniego drzewka? Skontaktuj się z nami!

E-mail: kontakt@bonsaiart.pl  
 Telefon: +48 123 456 789

Jesteśmy dostępny od poniedziałku do piątku, w godzinach 9:00 - 17:00. Chętnie pomożemy Ci na każdym etapie przygody z bonsai!

Kod:

```
import './ContactSection.scss'
import Container from "../../components/Container/Container";
import ContactForm from "../../components/ContactForm/ContactForm";
import SectionTitle from
"../../components/SectionTitle/SectionTitle";

const ContactSection = ({id}) => {
  return (
    <section id={id} className="ContactSection">
      <Container className="ContactSection__container">
        <ContactForm></ContactForm>
        <div className="ContactSection__content">
          <SectionTitle
            titlePart1="Masz jakieś pytania"
            titlePart2="Napisz do nas!"
          ></SectionTitle>
        </div>
      </Container>
    </section>
  );
}

export default ContactSection;
```

```

        <p
className="ContactSection__content__description">
    Chcesz dowiedzieć się więcej o pielęgnacji
bonsai? A może masz pytania dotyczące wyboru odpowiedniego drzewka?
Skontaktuj się z nami!
</p>
<p
className="ContactSection__content__description">
    ☎ E-mail: kontakt@bonsaiart.pl<br/>
    ☎ Telefon: +48 123 456 789
</p>
<p
className="ContactSection__content__description">
    Jesteśmy dostępni od poniedziałku do piątku, w
godzinach 9:00 - 17:00.[  
]SEP]Chętnie pomożemy Ci na każdym etapie przygody
z bonsai!
</p>
</div>
</Container>
</section>
)
};

export default ContactSection;

```

### Funkcja komponentu:

ContactSection: Sekcja umożliwiająca użytkownikom kontakt poprzez formularz oraz zawierająca dane kontaktowe.

### Propsy:

- id: Identyfikator sekcji, umożliwiający nawigację kotwicznej.

### Komponenty używane w ContactSection:

- Container: Zarządza szerokością i układem treści sekcji.
- ContactForm: Formularz kontaktowy do przesyłania zapytań.
- SectionTitle: Wyświetla tytuł sekcji w dwóch częściach.

### Struktura JSX:

- section.ContactSection: Główny kontener sekcji.
- Container.ContactSection\_\_container: Kontener zarządzający układem sekcji.
- div.ContactSection\_\_content: Kontener tekstowy zawierający tytuł i dane kontaktowe.

Style src/pages/Home/ContactSection/ContactSection.scss:

```
@import '../../styles/Global';

.ContactSection{
    width: 100%;
    padding: 7.5rem 0;
    font-size: 2rem;
    &__container{
        display: grid;
        grid-template-columns: 1fr 1fr;
        column-gap: 6rem;
        justify-items: center;
        @media screen and (max-width: $media-1024px) {
            grid-template-columns: 1fr;
        }
    }
    &__content{
        display: grid;
        row-gap: 2rem;
        @media screen and (max-width: $media-1024px) {
            grid-row: 1/2;
            justify-items: center;
            text-align: center;
            margin-bottom: 5rem;
        }
        &__description{
    }
}
```

```
}
```

### Główna klasa (.ContactSection)

- width: 100%: Sekcja zajmuje pełną szerokość strony.
- padding: 7.5rem 0: Dodaje odstęp nad i pod sekcją, zwiększając przestrzeń wizualną.
- font-size: 2rem: Ustawia podstawowy rozmiar tekstu dla sekcji.

### Kontener (.ContactSection\_\_container)

- display: grid: Organizuje treść w układzie siatki.
- grid-template-columns: 1fr 1fr: Dzieli sekcję na dwie kolumny o równych szerokościach.
- column-gap: 6rem: Dodaje odstępy między kolumnami.
- justify-items: center: Wyrównuje elementy siatki do środka.

#### Responsywność:

##### Do 1024px:

- grid-template-columns: 1fr: Zmienia układ na jednokolumnowy, aby lepiej dostosować się do mniejszych ekranów.

### Treść (.ContactSection\_\_content)

- display: grid: Organizuje elementy tekstowe w układzie siatki.
- row-gap: 2rem: Dodaje odstępy między wierszami.

#### Responsywność:

##### Do 1024px:

- grid-row: 1/2: Ustawia kolejność treści w pierwszym wierszu siatki.
- justify-items: center: Wyrównuje elementy do środka siatki.
- text-align: center: Centruje tekst w poziomie.
- margin-bottom: 5rem: Dodaje odstęp poniżej treści.

## Rodzaje src/pages/TypeOfBonsai/TypeOfBonsai.js

BONSAI.PL DLA POCZĄTKUJĄCYCH ▾ RODZAJE GALERIA KONTAKT 1585px 907px

### Rodzaje Drzewek Bonsai

Drzewka bonsai to wyjątkowe połączenie natury i sztuki. Ich miniatura forma, misterne kształtowana przez lata, stanowi nie tylko ozdobę, ale także wyraz cierpliwości i precyzji.



#### Ficus Acer



Opis: Popularny wybór dla początkujących dzięki dużej odporności na bardziej trudne warunki. Fikusy mają grubie, błyszczące liście i charakterystyczne korzenie powietrzne. Cechy szczególne: Wymagają ciepła i dobrze radzą sobie w warunkach domowych.



Opis: Popularny wybór dla początkujących dzięki dużej odporności na bardziej trudne warunki. Fikusy mają grubie, błyszczące liście i charakterystyczne korzenie powietrzne. Cechy szczególne: Wymagają ciepła i dobrze radzą sobie w warunkach domowych.

#### Juniperus Prunus



Opis: Popularne drzewko bonsai o gęstych gałęziach, co umożliwia kształtowanie ich w różnych style, np. kaskadowy. Ich igły lub łuski mają różne odcienie zielonych. Cechy szczególne: Są odporne i dobrze radzą sobie na zewnątrz, nawet w chłodniejszych warunkach.



Opis: Popularny wybór dla początkujących dzięki dużej odporności na bardziej trudne warunki. Fikusy mają grubie, błyszczące liście i charakterystyczne korzenie powietrzne. Cechy szczególne: Wymagają ciepła i dobrze radzą sobie w warunkach domowych.

#### Pinus Olea



Opis: Sosny są klasycznym wyborem w bonsai, często wybierane ze względu na ich surowy wygląd i wytrzymałość. Posiadają igły w różnych odcieniach zielonych. Cechy szczególne: Idealne na zewnątrz, sosny wymagają pełnego słońca i są odporne na bardziej trudne warunki atmosferyczne.



Opis: Drzewko o drobnym, srebrzystym liściach i interesującym, chropowatym pniu. Symbolizuje długowieczność i często stosowane jest jako bonsai na południu Europy. Cechy szczególne: Lubią dużo światła i jest odporna na suszę, idealna do cieplejszych klimatów.

Stworzone przez Norbert Fuk

Kod:

```
import './TypesSection.scss'  
import Container from ' ../../components/Container/Container'  
import SingleBonsai from "./SingleBonsai/SingleBonsai";  
import treeData from "../treeData";
```

```

const TypesSection = () =>{
  console.log(treeData);
  return (
    <section className="TypesSection">
      <Container className="TypesSection__container">
        {Object.values(treeData).map((tree, index) => {
          let style = '';
          if (index % 2 !== 0) {
            style = 'SingleBonsai__left';
          }
          return (
            <SingleBonsai
              key={index}
              img={tree.img}
              title={tree.title}
              description={tree.description}
              className={style} // Zakładam, że
`className` jest wspierany w `SingleBonsai`
            />
          );
        ))}
      </Container>
    </section>
  )
}

export default TypesSection;

```

### Funkcja komponentu:

TypeOfBonsai: Strona prezentująca różne rodzaje drzewek bonsai, składająca się z sekcji wprowadzającej (Hero) i głównej sekcji z typami bonsai (TypesSection).

### Dane statyczne (heroData):

- titlePart1: Pierwsza część tytułu wyświetlanego w sekcji Hero.
- titlePart2: Druga część tytułu wyświetlanego w sekcji Hero.
- description: Tekst opisowy w sekcji Hero, wprowadzający użytkownika w tematykę.

### Komponenty używane w TypeOfBonsai:

- Hero: Sekcja główna strony, zawierająca tytuł i opis.
- TypesSection: Główna sekcja strony wyświetlająca różne typy drzewek bonsai.

#### Efekty (useEffect):

- Ustawia tytuł strony na "Rodzaje - Bonsai.pl" przy załadowaniu komponentu.

#### Struktura JSX:

- main.typeOfBonsai: Główny kontener strony, zawierający wszystkie sekcje.

#### Style:

brak

## Komponenty:

Src/pages>TypeOfBonsai/TypesSection/TypesSection.js



### Ficus Acer

Opis: Popularne wybór dla początkujących dając dużej odporności na brudne warunki. Drzewy mają grube, wyciągnięte liście i charakterystyczne żółte i pomarańczowe liście wiosenne. Wymagały dużej i głębokiej doniczki w warunkach domowych.



Opis: Popularny wybór dla początkujących drzewek o odporności na zimne warunki. Drzewy mają grube, wyciągnięte liście i charakterystyczne żółte i pomarańczowe liście wiosenne. Cechy szczególnie wymagają ciepła i dobrze nadzorowanej warunków domowych.



### Juniperus Prunus

Opis: Popularne drzewka bonsai z gęstymi gałęziami, co stanowią idealne dla wiosennego stylu. Ma krótkie igły lub liść mała i cienka dekoracyjna. Cechy szczególne: są uprzednio zatrzymane i rosną na nowym, nawet w całkowicie zarośniętych warunkach.



Opis: Popularny wybór dla początkujących drzewek o odporności na zimne warunki. Drzewy mają grube, wyciągnięte liście i charakterystyczne korzenie powietrzne. Cechy szczególne: wymagają ciepła i dobrze nadzorowanej warunków domowych.



### Pinus Olea

Opis: Sosny ze klasycznym wyborem w formie, często wybierane ze względu na ich surowy wygląd i wyjątkowość. Posiadają głęboką i głęboko odciętą koronę. Cechy szczególne: idealne na zimowy, sosny wymagają pełnego słońca i są ostrożne na zmienne warunki atmosferyczne.



Opis: Drzewko o drobnich, smukłych liściach i interesującym, chropowatym pniu. Brakującą cezurę nadwodną i czekoladę stanowią jasne białe miękkie pączki bambusa. Cechy szczególne: małe i głębokie doniczki, jest odporna na mroz. Idealna do ciepłych klimatów.

## Kod:

```
import './TypesSection.scss'  
import Container from '../../../../../components/Container/Container'  
import SingleBonsai from "./SingleBonsai/SingleBonsai";  
import treeData from "../treeData";  
  
const TypesSection = () =>{  
    console.log(treeData);  
    return (  
        <section className="TypesSection">  
            <Container className="TypesSection__container">  
                {Object.values(treeData).map((tree, index) => {
```

```

        let style = '';
        if (index % 2 !== 0) {
            style = 'SingleBonsai_left';
        }
        return (
            <SingleBonsai
                key={index}
                img={tree.img}
                title={tree.title}
                description={tree.description}
                className={style} // Zakładam, że
`className` jest wspierany w `SingleBonsai`
            />
        );
    )})
</Container>
</section>
)
}

export default TypesSection;

```

### Funkcja komponentu:

TypesSection: Sekcja wyświetlająca listę różnych rodzajów bonsai w formie kart z informacjami o drzewkach.

### Dane (treeData):

- Obiekt treeData zawiera informacje o drzewkach bonsai, takie jak obraz (img), tytuł (title), oraz opis (description).

### Dynamiczne renderowanie:

- Używana jest metoda Object.values(treeData).map() do iteracji przez dane drzewek.

### Style:

- Klasa SingleBonsai\_left jest dodawana naprzemiennie do co drugiego elementu (dla indeksów nieparzystych), aby różnicować układ.

### **Komponenty używane w TypesSection:**

- Container: Opakowuje całą sekcję i zarządza szerokością treści.
- SingleBonsai: Reprezentuje pojedyncze drzewko bonsai jako kartę z obrazem, tytułem i opisem.

### **Struktura JSX:**

- section.TypesSection: Główny kontener sekcji.
- Container.TypesSection\_\_container: Kontener zarządzający układem kart bonsai.

Style src/pages>TypeOfBonsai/TypesSection/TypesSection.scss:

```
@import "../../styles/Global";
.TypesSection{
  padding: 7.5rem 0;
  @media screen and (max-width: $media-425px) and (min-width: 0){
    margin-top: 7rem;
  }
  &__container{
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: repeat(3, max-content);
    grid-gap: 5rem 3rem;

    @media screen and (max-width: $media-1440px) {
      grid-template-columns: 1fr;
      max-width: 70rem;
    }
    @media screen and (max-width: $media-768px) {
      grid-gap: 2rem 3rem;
    }
  }
}
```

### **Główna klasa (.TypesSection)**

- padding: 7.5rem 0: Dodaje odstępy nad i pod sekcją.

### **Responsywność:**

### **Do 425px:**

- margin-top: 7rem: Dodaje margines nad sekcją, aby uwzględnić układ mobilny.

### **Kontener (.TypesSection\_\_container)**

- display: grid: Ustawia elementy w układzie siatki.
- grid-template-columns: 1fr 1fr: Tworzy dwie kolumny o równych szerokościach.
- grid-template-rows: repeat(3, max-content): Tworzy trzy wiersze dopasowane do treści.
- grid-gap: 5rem 3rem: Dodaje odstępy między elementami w pionie i poziomie.

### **Responsywność:**

#### **Do 1440px:**

- grid-template-columns: 1fr: Zmienia układ na jednokolumnowy.
- max-width: 70rem: Ogranicza szerokość kontenera.

#### **Do 768px:**

- grid-gap: 2rem 3rem: Zmniejsza odstępy między elementami w układzie siatki.

*Src/pages>TypeOfBonsai/TypesSection/SingleBonsai/SingleBonsai.js*



## Ficus Acer

Opis: Popularny wybór dla początkujących dzięki dużej odporności na bardziej trudne warunki. Fikusy mają grubą, błyszczącą liście i charakterystyczne korzenie powietrzne. Cechy szczególne: Wymagają ciepła i dobrze radzą sobie w warunkach domowych.

Opis: Popularny wybór dla początkujących dzięki dużej odporności na bardziej trudne warunki. Fikusy mają grubą, błyszczącą liście i charakterystyczne korzenie powietrzne. Cechy szczególne: Wymagają ciepła i dobrze radzą sobie w warunkach domowych.



### **Kod:**

```
import './SingleBonsai.scss'

const SingleBonsai = ({className=' ',img,title,description}) => {
  return (
    <div className={`${SingleBonsai ${className}}`}>
```

```

        <img src={img} alt="Drzewko Bonsai Galeria"
      className="SingleBonsai__img"/>
        <div className="SingleBonsai__title">{title}</div>
        <div
      className="SingleBonsai__description">{description}</div>
        </div>
      )
}

export default SingleBonsai;

```

### Funkcja komponentu:

SingleBonsai: Wyświetla pojedyncze drzewko bonsai w formie karty z obrazem, tytułem i opisem. Umożliwia dynamiczne modyfikowanie układu za pomocą klasy className.

### Propsy:

- className: Opcjonalna klasa CSS, pozwalająca na dostosowanie układu (np. SingleBonsai\_left dla lewego wyrównania).
- img: URL obrazu reprezentującego drzewko bonsai.
- title: Tytuł drzewka bonsai.
- description: Opis drzewka bonsai.

### Struktura JSX:

- div.SingleBonsai: Główny kontener karty.
- img.SingleBonsai\_img: Obraz drzewka bonsai.
- div.SingleBonsai\_title: Tytuł drzewka bonsai.
- div.SingleBonsai\_description: Opis drzewka bonsai.

Style src/pages/TypeOfBonsai/TypesSection/SingleBonsai/SingleBonsai.scss:

```

@import "../../../../../styles/Global";
.SingleBonsai{
  //background: red;
  display: grid;
  text-align: right;
  grid-template-areas:
    'img title title'

```

```
'img description description';
align-items: start;
@media (max-width: $media-768px){
  grid-template-areas:
    'img'
    'title'
    'description' !important;
  justify-items: center;
  text-align: center;
}
&__left{
  grid-template-areas:
    'title title img'
    'description description img' !important;
  @media (max-width: $media-768px){
    grid-template-areas:
      'img'
      'title'
      'description' !important;
    text-align: center !important;
  }
  text-align: left !important;
}

&__img{
  grid-area: img;
  width: 25rem;
  height: 30rem;
  filter: drop-shadow(0 0 4px rgba(0,0,0,0.5));
  //background:gray;
}
&__title{
  font-size: 8rem;
  font-weight: 600;
  position: relative;
  height: 12.8rem;
  @media (max-width: $media-768px){
```

```

        font-size: 6.4rem;
    }
    @media (max-width: $media-425px){
        font-size: 4.8rem;
        height: 8rem;
    }
    & > *{
        position: absolute;

    }

    grid-area: title;
}
&__description{
    grid-area: description;
    display: flex;
    align-items: start;
    justify-items: start;
    height: 25rem;
    @media (max-width: $media-768px){
        height: 13rem;
    }
    @media (max-width: $media-425px){
        height: 25rem;
    }
}
}
}

```

## Główna klasa (.SingleBonsai)

### **Układ:**

- display: grid; Ustawia siatkę dla elementów karty.
- grid-template-areas: Określa układ elementów w siatce:
  - Domyślnie obraz jest po lewej, a tytuł i opis po prawej.
  - img title title oraz img description description.

### **Responsywność:**

#### **Do 768px:**

- Układ zmienia się na jednokolumnowy ('img' 'title' 'description').
- Elementy są wyśrodkowane (justify-items: center) i tekst jest centrowany (text-align: center).

### Lewa orientacja (.SingleBonsai\_left)

#### Układ:

- grid-template-areas: Zmienia orientację, przenosząc obraz na prawo ('title title img' oraz 'description description img').
- text-align: left: Wyrównuje tekst do lewej.

#### Responsywność:

##### Do 768px:

- Układ wraca do jednokolumnowego, jak w domyślnej konfiguracji.

### Obraz (.SingleBonsai\_img)

- grid-area: img: Przypisuje obraz do obszaru img w siatce.
- width: 25rem: Ustawia szerokość obrazu.
- height: 30rem: Ustawia wysokość obrazu.
- filter: drop-shadow: Dodaje cień wokół obrazu.

### Tytuł (.SingleBonsai\_title)

- grid-area: title: Przypisuje tytuł do obszaru title w siatce.
- font-size: 8rem: Duży rozmiar czcionki dla tytułu.
- font-weight: 600: Ustawia pogrubioną czcionkę.
- position: relative: Pozwala na pozycjonowanie wewnętrznych elementów.

#### Responsywność:

##### Do 768px:

- Zmniejsza rozmiar czcionki do 6.4rem.

##### Do 425px:

- Dalsze zmniejszenie rozmiaru czcionki do 4.8rem.
- Zmniejsza wysokość do 8rem.

#### **Opis (.SingleBonsai\_description)**

- grid-area: description: Przypisuje opis do obszaru description w siatce.
- display: flex: Używa układu elastycznego do wyrównania treści.
- align-items: start: Wyrównuje elementy w pionie do góry.
- height: 25rem: Ustawia wysokość opisu.

#### **Responsywność:**

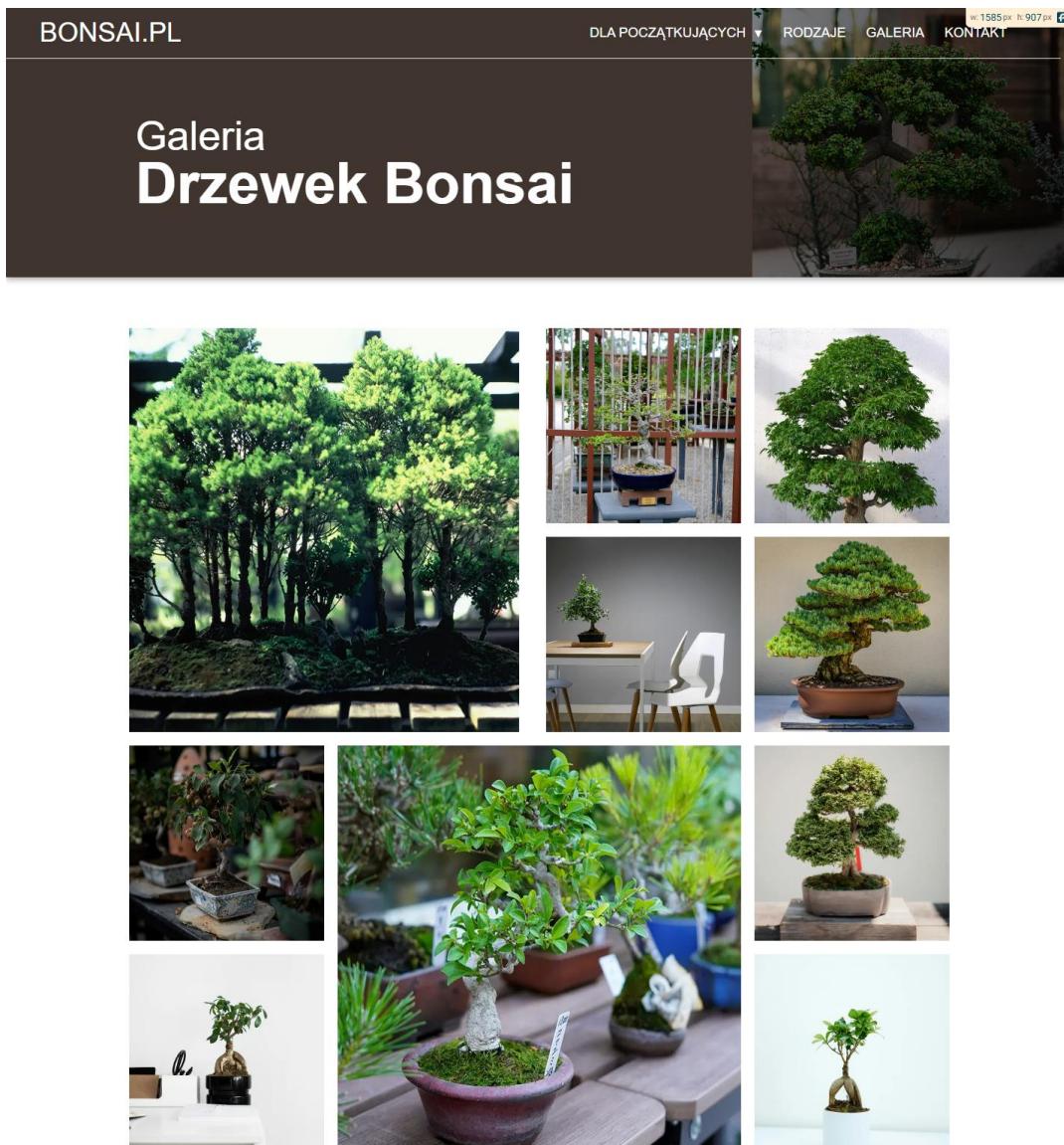
##### **Do 768px:**

- Zmniejsza wysokość do 13rem.

##### **Do 425px:**

- Wysokość pozostaje ustawiona na 25rem.

## Galeria src/pages/Gallery/Gallery.js



Stworzone przez Norbert Fuk

Kod:

```
import './Gallery.scss'  
import Container from "../../components/Container/Container";  
import Hero from "../../components/Hero/Hero";  
  
import photo1 from "../../assets/Gallery/arnaud-berthomier-SMbW_DyXGf4-unsplash.webp";
```

```

import photo2 from "../../assets/Gallery/carlos-gonzalez-31IcHku4g30-
unsplash.webp";
import photo3 from "../../assets/Gallery/dario-bertolino-MQIRaF7SEyc-
unsplash.webp";
import photo4 from "../../assets/Gallery/davide-cantelli-ajisKc2uuFk-
unsplash.webp";
import photo5 from "../../assets/Gallery/devin-h-PAEwrnas0vY-
unsplash.webp";
import photo6 from "../../assets/Gallery/jesus-arango-iOL0MsrtTtrU-
unsplash.webp";
import photo7 from "../../assets/Gallery/katy-t_Oxcbm12Vw-
unsplash.webp";
import photo8 from "../../assets/Gallery/mark-tegethoff-TYUS-cXzy50-
unsplash.webp";
import photo9 from "../../assets/Gallery/sarah-dorweiler-KcuFLkTXYy4-
unsplash.webp";
import photo10 from "../../assets/Gallery/toa-heftiba-W1yjvf5idqA-
unsplash.webp";
import {useEffect} from "react";
const heroData = {
    titlePart1: "Galeria",
    titlePart2: "Drzewek Bonsai",
};

const Gallery = () => {
    useEffect(() => {
        document.title = "Galeria - Bonsai.pl"; // Ustaw nazwę strony
    }, []);
    return (
        <main className="Gallery">
            <Hero
                titlePart1={heroData.titlePart1}
                titlePart2={heroData.titlePart2}
            />
            <Container className='Gallery__container'>
                <img src={photo1} className={'Gallery__photo
Gallery__photo--1'} alt={'test'}/>
                <img src={photo2} className={'Gallery__photo
Gallery__photo--2'} alt={'test'}/>

```

```

        <img src={photo3} className={'Gallery__photo
Gallery__photo--3'} alt={'test'}/>
        <img src={photo4} className={'Gallery__photo
Gallery__photo--4'} alt={'test'}/>
        <img src={photo5} className={'Gallery__photo
Gallery__photo--5'} alt={'test'}/>
        <img src={photo6} className={'Gallery__photo
Gallery__photo--6'} alt={'test'}/>
        <img src={photo7} className={'Gallery__photo
Gallery__photo--7'} alt={'test'}/>
        <img src={photo8} className={'Gallery__photo
Gallery__photo--8'} alt={'test'}/>
        <img src={photo9} className={'Gallery__photo
Gallery__photo--9'} alt={'test'}/>
        <img src={photo10} className={'Gallery__photo
Gallery__photo--10'} alt={'test'}/>
    </Container>
</main>
);
}

export default Gallery;

```

### Funkcja komponentu:

Gallery: Strona przedstawiająca galerię drzewek bonsai w układzie siatki.

### Dane (heroData):

- titlePart1: Pierwsza część tytułu wyświetlanego w sekcji Hero.
- titlePart2: Druga część tytułu wyświetlanego w sekcji Hero.

### Obrazy w galerii:

- Importowane zdjęcia bonsai są wyświetlane w siatce o dynamicznym układzie.

### Komponenty używane w Gallery:

- Hero: Sekcja wprowadzająca z tytułem galerii.
- Container: Kontener zarządzający szerokością i układem siatki galerii.

### Efekty (useEffect):

- Ustawia tytuł strony na "Galeria - Bonsai.pl" przy załadowaniu komponentu.

#### Struktura JSX:

- main.Gallery: Główny kontener strony, zawierający wszystkie sekcje.
- Container.Gallery\_\_container: Kontener zarządzający układem zdjęć w siatce.

Style src/pages/Gallery/Gallery.scss:

```
@import "../../styles/Global";
.Gallery{
  &__container{
    padding: 7.5rem 0;
    display: grid;
    grid-template-areas:
      'photo1 photo1 photo2 photo3'
      'photo1 photo1 photo4 photo5'
      'photo6 photo7 photo7 photo8'
      'photo9 photo7 photo7 photo10';
    grid-gap: 2rem;
    justify-content: center;
    align-items: center;
    @media screen and (max-width: $media-1024px) and (min-width: $media-768px){
      margin-top: 7rem;
    }

    @media screen and (max-width: $media-1440px){
      grid-template-areas:
        'photo1 photo1 photo2'
        'photo1 photo1 photo3'
        'photo4 photo7 photo7'
        'photo5 photo7 photo7'
        'photo6 photo8 photo9'
    }

    @media screen and (max-width: $media-1024px){
      grid-template-areas:
        'photo1 photo2'
```

```

'photo3 photo4'
'photo5 photo6'
'photo7 photo8'
'photo9 photo10'
}

@media screen and (max-width: $media-768px){
  grid-template-areas:
    'photo1'
    'photo2'
    'photo3'
    'photo4'
    'photo5'
    'photo6'
    'photo7'
    'photo8'
    'photo9'
    'photo10'
}
}

&__photo{
  //width: 40rem;
  &--1, &--7{
    width: 60rem;
    height: 60rem;
  }
  &--2,&--3,&--4,&--5,&--6,&--9,&--8,&--10{
    width: 29rem;
    height: 29rem;
  }
  @media screen and (max-width: $media-1024px){
    width: 29rem !important;
    height: 29rem !important;
  }
  @media screen and (max-width: $media-768px){
    width: 35rem !important;
    height: 35rem !important;
  }
}

```

```
}

@media screen and (max-width: $media-375px){
    width: 28rem !important;
    height: 28rem !important;
}

&--1{
    grid-area: photo1;
    width: 58rem;
}
&--2{
    grid-area: photo2;
}
&--3{
    grid-area: photo3;
}
&--4{
    grid-area: photo4;
}
&--5{
    grid-area: photo5;
}
&--6{
    grid-area: photo6;
}
&--7{
    grid-area: photo7;
}
&--8{
    grid-area: photo8;
}
&--9{
    grid-area: photo9;
}
&--10{
    grid-area: photo10;
    @media screen and (max-width: $media-1440px){
        display: none;
    }
}
```

```
    @media screen and (max-width: $media-1024px){
      display: block;
    }
  }
}
```

### Główna klasa (.Gallery)

Cel: Służy jako główny kontener galerii.

### Kontener siatki (.Gallery\_\_container)

- padding: 7.5rem 0: Dodaje odstępy nad i pod siatką.
- display: grid: Ustawia elementy w układzie siatki.
- grid-template-areas: Definiuje układ zdjęć w siatce z nazwanymi obszarami.
- grid-gap: 2rem: Dodaje odstępy między zdjęciami.

#### Responsywność:

##### Do 1440px:

- Układ siatki zmienia się, aby dostosować się do mniejszej szerokości ekranu.

##### Do 1024px:

- Siatka przekształca się w dwukolumnową strukturę.

##### Do 768px:

- Zdjęcia są wyświetlane w jednej kolumnie w układzie pionowym.

### Zdjęcia (.Gallery\_\_photo)

#### Rozmiary:

width i height: Ustawiają wymiary zdjęć, różne dla zdjęć głównych i pobocznych.

##### Do 1024px:

- Zdjęcia mają jednakowe wymiary (29rem x 29rem).

##### Do 768px:

- Zdjęcia mają większe wymiary (35rem x 35rem).

#### **Do 375px:**

- Zdjęcia zmniejszają się (28rem x 28rem).

#### **Zdjęcia według obszaru siatki:**

- grid-area: Każde zdjęcie jest przypisane do zdefiniowanego obszaru (photo1, photo2, itd.).

#### **Specyficzne zachowania:**

##### **Gallery\_photo--10:**

###### **Do 1440px:**

- Ukrywane na ekranach o szerokości mniejszej niż 1440px.

###### **Do 1024px:**

- Wyświetlane ponownie.

## **Podsumowanie**

Dokumentacja opisuje kompletną strukturę, funkcjonalność oraz stylizację aplikacji internetowej prezentującej drzewko bonsai. Uwzględnia szczegóły implementacji komponentów, takich jak strony, sekcje, dynamiczne elementy interfejsu oraz style CSS/SCSS, a także wyjaśnia ich przeznaczenie i działanie